MODELING AND SIMULATION OF SINGLE AND MULTIPLE CLUSTER FRACTALS CULTURED IN POLYMER FILMS

SHAHIZAT BIN AMIR

THESIS SUBMITTED IN FULFILLMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

INSTITUTE OF GRADUATE STUDIES
UNIVERSITY OF MALAYA
KUALA LUMPUR

2013

UNIVERSITI MALAYA

PERAKUAN KEASLIAN PENULISAN

Nama: Shahizat Bin Amir (No. K.P/Pasport: 790614-14-6135)

No. Pendaftaran/Matrik: HHC090001

Nama Ijazah: Doktor Falsafah

Tajuk Kertas Projek/Laporan Penyelidikan/Disertasi/Tesis ("Hasil Kerja ini"):

MODELING and SIMULATION of SINGLE and MULTIPLE CLUSTER FRACTALS in POLYMER FILMS

Bidang Penyelidikan: Matematik Gunaan

Saya dengan sesungguhnya dan sebenarnya mengaku bahawa:

(1) Saya adalah satu-satunya pengarang/penulis Hasil Kerja ini;
(2) Hasil Kerja ini adalah asli;
(3) Apa-apa penggunaan mana-mana hasil kerja yang mengandungi hakcipta telah dilakukan secara urusan yang wajar dan bagi maksud yang dibenarkan dan apa-apa petikan, ekstrak, rujukan atau pengeluaran semula daripada atau kepada mana-mana hasil kerja yang mengandungi hakcipta telah dinyatakan dengan sejelasnya dan secukupnya dan satu pengiktirafan tajuk hasil kerja tersebut dan pengarang/penulisnya telah dilakukan di dalam Hasil Kerja ini;
(4) Saya tidak mempunyai apa-apa pengetahuan sebenar atau patut semunasabahnya tahu bahawa penghasilan Hasil Kerja ini melanggar suatu hakcipta hasil kerja yang lain;
(5) Saya dengan ini menyerahkan kesemua dan tiap-tiap hak yang terkandung di dalam hakcipta Hasil Kerja ini kepada Universiti Malaya ("UM") yang seterusnya mula dari sekarang adalah tuan punya kepada hakcipta di dalam Hasil Kerja ini dan apa-apa pengeluaran semula atau penggunaan dalam apa jua bentuk atau dengan apa juga cara sekalipun adalah dilarang tanpa terlebih dahulu mendapat kebenaran bertulis dari UM;
(6) Saya sedar sepenuhnya sekiranya dalam masa penghasilan Hasil Kerja ini saya telah melanggar suatu hakcipta hasil kerja yang lain sama ada dengan niat atau sebaliknya, saya boleh dikenakan tindakan undang-undang atau apa-apa tindakan lain sebagaimana yang diputuskan oleh UM.

Tandatangan Calon Tarikh 7/06/2013

Diperbuat dan sesungguhnya diakui di hadapan,

Tandatangan Saksi Tarikh 7/06/2013

Nama: PROF. DR. NOR SABIRIN MOHAMED
Jawatan: Bahagian Fizik
Pusat Asasi Sains
Universiti Malaya

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Shahizat Bin Amir (I.C/Passport No: 790614146135)

Registration/Matric No: HHC090001

Name of Degree: Doctor of Phylosophy

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

MODELING and SIMULATION of SINGLE and MULTIPLE CLUSTER FRACTALS in POLYMER FILMS

Field of Study: Applied Mathematics

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date 7/06/2013

Subscribed and solemnly declared before,

Witness's Signature Date 7/06/2013

Name: PROF. DR. NOR SABIRIN MOHAMED
Designation: Bahagian Fizik
Pusat Asasi Sains
Universiti Malaya

ABSTRACT

Study of fractals has been an interest for scientists and mathematicians since the term ‘fractal’ was first coined by Mandelbrot. By a simple definition, a fractal is a shape made of parts similar to the whole in some way. Many studies on fractals have been carried out either in applications, usually involving experimental works, or in theory, where most simulations on fractal patterns models are on nature-based fractals such as river flows, coastline and tree branching. Many of fractal growth models are suitable with experimental phenomena such as electrochemical electrodeposition, electrochemical polymerization and Diffusion Limited Aggregation growth structures of many metal aggregates in the presence of a magnetic field as external stimuli. The formation of fractals without using any external stimuli has been reported by a few groups of researchers. This work focuses on the improvisation of the modeling and simulation of laboratory cultured fractals using polymer electrolyte films as the media of growth. This research work’s main topics are fractals and fractal growth models particularly DLA (Diffusion Limited Aggregation). DLA model describes how a fractal is built from particles in low concentrations. The DLA cluster formed through DLA is formed by particles moving due to Brownian motion (diffusion) which meet and stick together randomly (aggregation) to form the cluster. Fractals can be constructed using this model from polymer films infused with inorganic salt without any external stimuli. The experimental methods for growing fractals are discussed. This research work also includes descriptions of polymer films properties and the advantages of using polymer

films infused with inorganic salt as media to culture fractals. The simulation of single and multiple cluster fractals is done using DLA methods incorporating different parameters such as its sticking coefficient, lattice geometry and number of particles. Development of a computer coding to simulate and visualize the fractal growth is a key part in this research. To compare the simulation with the real patterns obtained, one vital aspect would be the calculation of their fractal dimension values. The computer program developed is able to calculate the fractal dimension value of each of the simulated fractal patterns. Suitable fractal dimension calculation method is employed according to its usefulness and efficiency. Fractal growth modeling and simulation such as done here can contribute to the understanding of other related studies concerning fractal growth found in areas including medical (nervous systems, cancer growth and more).

ABSTRAK

Kajian fraktal telah menjadi satu kecenderungan untuk ahli-ahli sains dan ahli-ahli matematik sejak istilah 'fractal' adalah terlebih dahulu dicipta oleh Mandelbrot. Dengan satu takrifan yang mudah, fraktal ialah sebuah bentuk diperbuat daripada bahagian-bahagian sebagaimana keseluruhan dalam beberapa cara. Banyak kajian ke atas fraktal telah dijalankan sama ada dalam aplikasi, biasanya melibatkan kerja-kerja eksperimen, atau pada teori , di mana kebanyakan simulasi-simulasi membuahkan corak fraktal berasaskan sifat seperti sungai mengalir, pencabangan garis pantai dan pokok. Kebanyakan daripada model-model pertumbuhan fraktal sesuai dengan cerapan fenomena eksperimen seperti elektroenapan elektrokimia, pempolimeran elektrokimia dan struktur-struktur pertumbuhan Diffusion Limited Aggregation (DLA) banyak logam mengumpulkan dalam kehadiran satu medan magnet sebagai rangsangan luaran. Pembentukan fraktal tanpa menggunakan mana-mana rangsangan luaran telah dilaporkan oleh beberapa kumpulan penyelidik. Kerja ini menumpukan pada menambahbaikan pemodelan dan simulasi fraktal secara pengkulturan makmal menggunakan filem elektrolit polimer sebagai media pertumbuhan. Topik utama kerja penyelidikan ini ialah fraktal dan model-model pertumbuhan fraktal seperti DLA. Model DLA menggambarkan bagaimana satu fraktal dibina dari zarah-zarah dalam kepekatan yang rendah. Kelompok DLA membentuk melalui DLA ditubuhkan oleh zarah-zarah yang bergerak disebabkan pergerakan Brown (resapan) yang mana bertemu dan berkumpul bersama-sama secara rawak (pengagregatan) membentuk kelompok. Fraktal boleh dibina menggunakan model ini dari filem-filem polimer dicetus dengan garam tak organik tanpa mana-mana rangsangan luaran. Cara-cara eksperimental untuk

mengkulturkan fraktal diperbincangkan. Kerja penyelidikan ini juga termasuk huraian-huraian filem polimer ciri-ciri dan kelebihan menggunakan filem-filem polimer diseduh dengan garam tak organik sebagai media untuk mengkulturkan fraktal. Simulasi tunggal dan berbilang kelompok fraktal dijalankan menggunakan kaedah-kaedah DLA menggabungkan parameter yang berbeza seperti pekali lekatannya, geometri kekisi dan jumlah zarah. Pembangunan pengekodan komputer mensimulasi dan membayangkan pertumbuhan fraktal ialah satu bahagian utama dalam penyelidikan ini. Untuk bandingkan simulasi dengan corak-corak sebenar yang diperolehi, satu aspek amat penting adalah pengiraan nilai-nilai dimensi fraktal. Program komputer maju juga akan dapat menghitung nilai dimensi fraktal setiap corak-corak fraktal tersimulasi. Kaedah pengiraan dimensi fraktal sesuai akan diambil kira menurut kegunaan dan kecekapannya. Pemodelan pertumbuhan fraktal dan simulasi seperti dilakukan dalam kajian ini boleh menyumbang bagi pemahaman kajian fraktal yang lain berkenaan pertumbuhan fraktal yang mungkin wujud di bidang seperti perubatan (sistem saraf, pertumbuhan kanser dan lain-lain).

ACKNOWLEDGEMENTS

I would like to sincerely express my gratitude especially a great thank you to my family-mother and brother, without whose advice and support I would not be here and for their patience and understanding notwithstanding all the troubles encounter during the completion of this thesis. I am grateful to many people and institutions particularly the Institute of Graduate Studies for their help in the preparation of this thesis. For the opportunity to work on this fascinating topic, the support, and the technical input, a special thank you to my supervisors Professor Dr. Nor Sabirin Mohamed and Associate Professor Dr. Siti Aishah Hashim Ali. Their guidance and supervision are very valuable and highly appreciated.

I am also grateful to my colleagues for data collection and programming support. I would also like to acknowledge with thanks the financial support by the University of Malaya Postgraduate Research Fund. Not forgetting the support from my fellow staffs from the Centre of Foundation Studies in Science. For further financial and - more importantly - personal support, I again wish to thank my mother and father and my whole family and friends.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF APPENDICES

LIST OF RECENT PUBLICATIONS

Papers presented in conference/seminars:

[1] **S. Amir**, N.S. Mohamed and S.A. Hashim Ali (2009), *Using Polymer Electrolyte Membranes as Media to Culture Fractals: A Simulation Study*, The International Conference on Functionalized and Sensing materials (FuSeM 2009), 07 Dec 2009 to 09 Dec 2009, Silpakorn University, Dept Mat Sci & Engn, Advanced Materials Research, 93-94, 35-38.

[2] **S. Amir**, S.A. Hashim Ali and N.S. Mohamed (2009). *A DLA Model Approach in the simulation of fractals in PVDF-HFP/PEMA-NH4CF3SO3-Cr2O3 nanocomposite polymer electrolyte membranes.* Proceedings of Malaysian Polymer International Conference, 21 Oct 2009 to 22 Oct 2009, Universiti Kebangsaan Malaysia.

Papers published in ISI-cited journals:

[1] **S. Amir**, N.S. Mohamed, and S. A. Hashim Ali, (2010). *Simulation Model of the Fractal Patterns in Ionic Conducting Polymer Films.* Central European Journal of Physics 8(1):150-158. IF 0.696 Q3

[2] **S. Amir**, S.A. Hashim Ali, and N.S. Mohamed (2011*). Studies of fractal growth patterns in poly (ethylene oxide) and chitosan membranes.* Ionics 17(2):121-125 IF 1.288 Q3

[3] **S. Amir**, S.A. Hashim_Ali, and N.S. Mohamed, (2012). *Implementation of a DLA Model in the Simulation of Fractals in PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 Nanocomposite Polymer Electrolyte Films*, Physica Scripta 84:045802. IF 1.204 Q2

LIST OF RESEARCH AWARDS AND PATENTS

[1] **Gold medal** in Ekspo Penyelidikan, Rekacipta & Inovasi 2007, 26-28 July 2007, Universiti Malaya, Kuala Lumpur.
Assoc. Prof. Dr. Siti Aishah Hashim Ali, Assoc. Prof. Dr. Nor Sabirin Mohamed and Shahizat Amir, for research project on *Simulation of Fractal*.

[2] **Bronze medal** in PECIPTA 2009, 8 - 10 October 2009, Kuala Lumpur Convention Centre (KLCC).
Assoc. Prof. Dr. Siti Aishah Hashim Ali, Assoc. Prof. Dr. Nor Sabirin Mohamed and Shahizat Amir, for research project on *Simulation Model of Fractal Growth Pattern*.

[1] Simulation Model of Fractal Growth Patterns, Patent, PIC/P/473/11/UM/SA/SM, 2011, (University)

[2] Medical Digital Image Data Processing, Patent, PIC/P/474//11/UM/SA/FD, 2011, (University)

C H A P T E R 1

INTRODUCTION

Study of fractals has been an interest for scientists and mathematicians since the term ‘fractal’ was first coined by Mandelbrot Mandelbrot (1983). By a simple definition, a fractal is a shape made of parts similar to the whole in some way. There is a sense of curiosity about fractals because they cannot be described by classical geometry since they are irregular. Fractals also exhibit interesting properties that can be used for variety of applications. Many studies on fractal have been carried out either in applications, usually involving experimental works, or in theory, where most simulations on fractal patterns models are on nature-based fractals such as river flows, coastlines (Richardson, 1961) and tree branching (Hastings and Sugihara, 1993).

The process by which a fractal can be grown from a solution is called Diffusion Limited Aggregation (DLA). In DLA, diffusion and aggregation are the two processes involved in forming a fractal. The solution must have a very low concentration of particles in order for a fractal to grow. The particles in the solution move around in random direction (Brownian motion) and they can stick together slowly forming a cluster (Brunner et al., 1995). Consequently many researchers study the growth and shapes of fractals through theoretical modeling and computer simulations of fractal patterns.

This work describes a simulation model of fractal patterns found in ionic conducting polymer films. The characteristics and suitability of the model have been studied and a computer program to simulate the growth of the pattern was developed.

1.1 Statement of Problem

There are three basic models of fractal growth: percolation (Bunde and Havlin, 1991), particle-cluster aggregation (PCA)(Vicsek, 1992, Tan et al., 1999, Meakin, 1983) and cluster-cluster aggregation (CCA) (Tan et al., 2000, Zhang and Liu, 1998). Among the models of PCA, DLA (Witten and Sander, 1983) is the most well-known. DLA has been used to describe diffusive systems including viscous fingering, electrochemical deposition, dielectric breakdown and monolayer formation on surface (Matsushita et al., 1984b, Paterson, 1984, Irurzun et al., 2002, He and Huang, 2008). Some progresses have been made in the description of fractal growth patterns in recent years using fractal geometry (Tan et al., 1999). Fractal geometry provides a new method to study the phenomena of fractal growth patterns under certain circumstances. However, up to now, most of the studies about fractal growth patterns have been limited to the calculation of the fractal dimension or the simulation by statistical models (Paterson, 1984, Akuezue and Stringer, 1989, March, 1992, Hentschel, 1992, Mukherjee et al., 1995, Praud and Swinney, 2005, Knudsen et al., 2008). Although the results provide some new understanding about the complexity of fractal growth patterns, a shortcoming is that these investigations concentrated only on geometry description. Therefore, details regarding the fractal geometry of such fractal growth pattern need be investigated further in order to get a clearer interpretation of any fractal growth phenomena. For that purpose, integration of three components: experimental, modeling and simulation study of fractal growth pattern will be the key elements in understanding the fractal growth pattern better.

Studies on the growth of dendrites of fractal pattern in a conducting polymer (Shui et al., 2004) have been done. Yet its application in secondary battery (Rosso, 2007, Mandelbrot,

1983) has not been fully understood. It is difficult to actually study directly the growth of dendrites of fractal pattern that forms in the electrode since the fractal patterns could be easily damaged during accumulation. Thus as an alternative, fractals can be cultured in ion conducting polymer membrane to replicate the condition in a similar environment via laboratory experiments. In this research, the main focus is to get a more effective DLA model by implementing an extension of the basic DLA model for the morphological evaluations of the fractal growth patterns. Furthermore, fractal growth pattern in nature or experimentally obtained, usually does not consist of only a single cluster, but multiple clusters as can be seen in polymer-salt membranes. Thus it is important to study the effects of neighboring clusters to the overall fractal growth pattern in such polymer membranes. This is investigated in this research work.

1.2 Research Background

The term ‘fractal’ refers to a family of complex geometrical shapes that can be characterized by a fractional or non-integer dimensionality and was introduced by Benoit B. Mandelbrot (1983). The concept of fractals has attracted the interest of scientists in many fields (Feder, 1988). A huge number of papers related to the word ‘fractal’ has been published, spanning fields ranging from physical geometry, such as surface structure of sea beds (Golubev et al., 1987), non-equilibrium growth phenomena (Shibkov et al., 2001) and distribution of intervals between earthquakes (Dargahi-Noubary, 1997), to ecology that involves fungal structure (Tordoff et al., 2008) and power law relationship between the area of a quadrate and the structure of peat systems (Sławiński et al., 2002). Works on fractals are also common in cosmology including the study of the structure of star clusters and galaxies, the big bang theory of the origin of the universe and also in developmental biology portrayed by lung branching patterns, heart rhythms and structure

of neurons (Hastings and Sugihara, 1993). The most amazing thing about fractal is the variety of its applications. Besides theoretical applications, fractals can be found in almost every part of the universe, from bacteria cultures to galaxies and to the human body. Many studies of fractals related to the field of astronomy (Combes, 1998), biology (Stanley et al., 1994) and chemistry (Villani and Comenges, 2000) have also been reported in the literature.

In mathematics, the study of fractals revolves around areas such as data compression, fractal art and diffusion. Many of fractal growth mathematical models were found to be suitable with experimental studies of electrochemical electrodeposition (Barkey, 1991), electrochemical polymerization (Kaufman et al., 1987), thin films (Catalan et al., 2008) and DLA growth structures of many metal aggregates in the presence of a magnetic field as external stimuli (Okubo et al., 1993). The formation of fractals without using any external stimuli has been done by a few groups of researchers (Chandra and Chandra, 1994, Chandra, 1996, Mohamed and Arof, 2001, Amir et al., 2010). Recent studies of fractals in polymers that involved modeling and/or simulation include those reported by Janke and Schakel (2005), Lo Verso et al. (2006) , Marcone et al. (2007) and Sorensen (2011). On the other hand, Rathgeber et al. (2006) have done some work on theoretical modeling and experimental studies of dendrimers. There have also been experimental studies of crystal pattern transition from dendrites through fourfold-symmetric structures to faceted crystals of ultra-thin poly (ethylene oxide) films which were carried out by Zhang et al. (2008). These research works on fractals were done only on laboratory experiments, theoretical modeling and experimental studies, or modeling and computer simulations. Recently, integration of all the three approaches; experimental, modeling and simulation have been reported (Amir et al., 2010, Amir et al., 2011). However, these works were only concentrated on the study of single cluster fractal growth patterns

without the inclusion of other fractal growth parameters such as sticking coefficients and different lattice sites. In the present research, attention is given on finding the passe-partout of the study of such fractal growth patterns by addition of other fractal growth parameters carried out not just for single cluster but also multiple clusters of fractal growth patterns. Further understanding on the formation of such aggregates can be achieved with the introduction of the fractal growth parameters mentioned above.

In the formation of fractal patterns, aggregation of particles is an important aspect. In a typical aggregation process, particles may escape from a cluster and undergo a random walk until they again reach the cluster or another cluster (Meakin, 1988). In essence, fractal dimension is an indicator of the aggregate structure, which also indirectly provides information about the strength of the aggregates (Gregory, 1998). In cluster growth models, a cluster gradually expands in its medium. The cluster is given some initial shape, and expansion occurs based on an aggregation algorithm. Simple algorithms often generate complex structures that resemble certain types of morphologies. Witten and Sander (1981) proposed a cluster growth model called DLA that simulates diffusion using random movements of particles. In this model, particles are assumed to move randomly through a two-dimensional grid until they collide with and stick to a growing aggregate. Surprisingly, this simple process generates complex branching structures with fractal dimension. Witten and Sander (Witten and Sander, 1981, Witten and Sander, 1983) have shown that the probability distribution of a DLA random walker follows Laplace's equation, which explains the complex patterns that these simulations produce. Kaandorp (1994) used an accretive growth model to simulate three-dimensional formation of corals and sponges. In this iterative model, layers of materials are added to a growing tip. The thickness of the layer can be parameterized such that more growth occurs at the tip than

along the sides. If this process is tuned properly, it can result in branching patterns that resemble corals and sponges.

The DLA method has been applied specifically to the Saffman-Taylor instability and the equations of Laplacian growth. Liang (1986) solved these equations using two types of random walkers. The first type originates far from the cluster and the second type originates at one of the boundary sites, chosen at random proportional to the local curvature. Meakin et al. (1987) used an off-lattice version of DLA together with a sticking probability based on the local curvature of the cluster to solve the same equations.

With that consideration, for this work, the characteristics and features of the model were studied and based on this model a computer program to simulate the growth patterns of aggregates cultured in polymer membranes was developed. In the process of developing the model, studies were carried out on related issues such as the fractal growth patterns and mechanism, and characteristics of the fractals. Some mathematical and computer modeling techniques associated with simulation model chosen were identified and implemented in the development of the computer programming system. Aspects that may influence the type and characteristics of the fractals formed shall be studied and taken into account.

1.3 Objectives and Scope of the Present Work

The main aim of the present work is to experimentally culture fractals using polymer electrolyte films and to simulate their growth patterns. The work for the present study was carried out with the objectives as follows:

(i) to obtain fractals using polymer electrolyte films as media of growth without any external stimuli

(ii) to observe and study the fractals formation and its' growth patterns and mechanism in the chosen (polymer - inorganic salt) systems

(iii) to perform fractal analysis of the cultured and simulated fractal growth patterns by calculating their fractal dimension values.

(iv) to develop computer programs for the generation of fractal patterns obtained in the chosen polymer electrolyte films.

(v) to achieve a better understanding of fractal growth processes by implementing simulation models of the single and multiple clusters of fractal growth patterns.

(vi) to investigate the effects of various fractal growth parameters on fractal growth patterns

1.4 Thesis Organization

This thesis consists of seven chapters. Chapter 1 begins with an introduction to the idea of how natural forms and patterns are viewed from the perspective of science and mathematics which is not always true when describing a certain pattern known as fractal. This chapter also includes the research background, the objectives and the scope of the present work.

The literature review on fractal background which includes the significance of fractals, fractal geometry and fractal mathematics are given in Chapter 2. Focus is given on the importance of fractals and their contributions to human life. The concept of fractal studies and theories involved are also included.

Chapter 3 focuses on the experimental work for growing fractals and the suitable simulation methods for the fractals. Fine details of fractals simulation computer program

development is explicated in this chapter. This chapter also explains how the determination of fractal dimension was done for both the experimentally obtained and simulated fractals.

Chapter 4 presents and discusses the results of the experimentally cultured fractal growth patterns observed in different types of polymer membranes which are gathered from the laboratory. Meanwhile, Chapter 5 addresses the substantiation of similar fractal growth patterns in achieving simulation model that best suit the experimentally cultured fractal.

Chapter 6 covers the results and discussions of the simulation works with due concentration to single cluster and multiple clusters fractal growth patterns. The conclusion of the present work and issues for further research in this area are presented in Chapter 7.

C H A P T E R 2

FRACTALS AND FRACTAL GROWTH MODELS

Since the 1970s many of nature's patterns have been shown to take the forms of fractal (Mandelbrot, 1983). In contrast to the smoothness of artificial lines, fractals consist of patterns that recur on finer and finer scales, building up shapes of immense complexity.

2.1 Evolution of Fractal

In order to understand and appreciate the development of research in fractal geometry, it is important to review the historic evolution of it, since fractal geometry was founded as the result of works done by many mathematicians centuries ago. As mentioned in Chapter 1, the word ‘fractal’ was first introduced in 1975 (Mandelbrot, 1983). Before this date, some fractal-like patterns have been observed but were generally described as ‘mathematical monsters’. The evolution of fractal geometry is summarized as follows:

1500: The first fractal drawing was painted by Albretch Dürer, an artist during the ‘Renaissance’. This picture was created with a main pentagon in which 5 similar pentagons were drawn (Durer, 1525). By repeating this operation over and over, the artist produced a picture as shown in Figure 2.1.

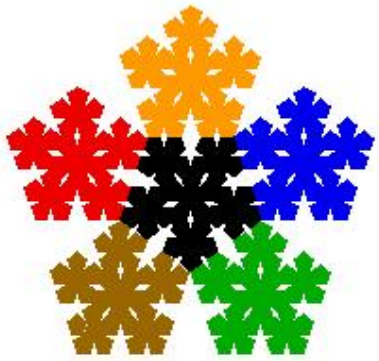


Figure 2.1: The Dürer's pentagon

1700: Gottfried Wilhelm Leibniz discovered the property of 'self-similarity' of some objects (Falconer, 2003). This discovery was then followed by the development of the differentiable functions by Newton and Leibniz (Barnsley and Hawley, 1993).

1883: Work on the Cantor's set (Peitgen et al., 2004), one of the oldest fractal geometry described as illustrated in Figure 2.2 was published. In order to obtain this object, $\frac{1}{3}$ of a central line of undefined length was taken out, and the process is repeated in each iteration.

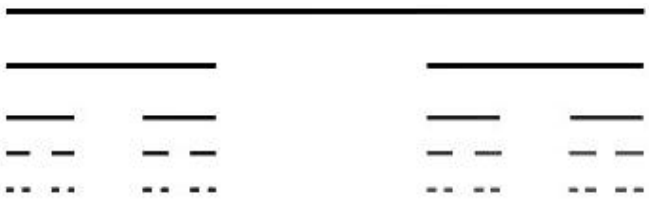


Figure 2.2: Representation of the Cantor's set after 4 iterations

1890: An Italian mathematician Giuseppe Peano defined a curve with several strange properties, which was called *monstrous curve*. It is a line (and therefore appears to be one-dimensional), but it fills a square (in the sense that it goes through every point in the square) and therefore could be considered two-dimensional. Another curious property of this curve is that it has no tangent or derivative at any point (Alfonseca and Ortega, 2001). This curve is obtained by the reiteration of a simple geometric operation on a line as illustrated in Figure 2.3.

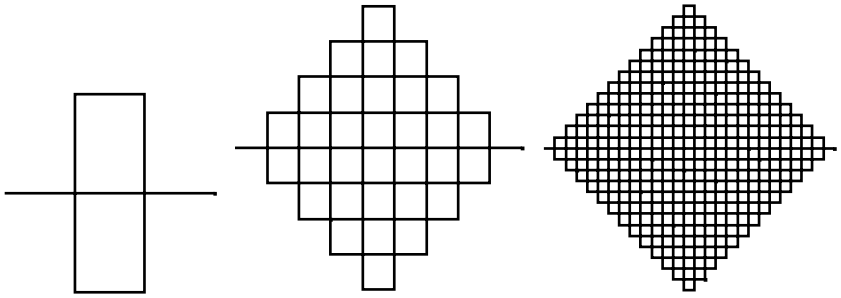


Figure 2.3: The Péano curve after 3 iterations

1904: Helge von Koch devised the Von Koch's snowflakes which like the Peano curve, has no derivative at any point, and its longitude is infinite, even though its size is limited (Alfonseca and Ortega, 2001). Its dimension seems to be larger than 1, although it does not fill the plane, and thus cannot reach 2. Figure 2.4 illustrates the Von Koch's snowflake. The structure after each iteration of this Von Koch's snowflake is shown in Figure 2.8

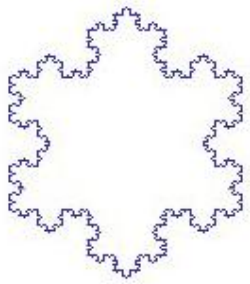


Figure 2.4: The Von Koch's snowflake

1915: The Sierpinski gasket was introduced (Barnsley and Hawley, 1993). This fractal image has been very popular, compared to the previous fractal images because the geometric operation is applied on a surface rather than on a line. It is also called Sierpinski's triangle, that is a new kind of fractal. In order to generate the Sierpiñski's triangle, a full equilateral triangle is drawn. Then a smaller equilateral triangle is taken out from the object. From the 3 triangles remaining, 3 smaller triangles are taken out as shown in Figure 2.5.

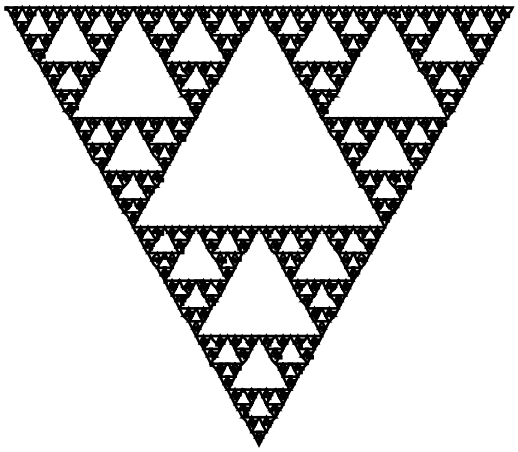


Figure 2.5: The Sierpiñski's triangle

1918: Gaston Maurice Julia, at the age of 25 published an article titled "*Mémoire sur l'itération des fonctions rationnelles*" in which he developed the concept of reiteration of polynomial functions (Falconer, 2003).

1919: Non-integer dimension was discovered by Felix Hausdorff, a German mathematician who is considered to be one of the founders of modern topology (Kahane, 1983). Hausdorff showed another way of measuring the dimensional aspect of a fractal object. His work helped Mandelbrot to give non-integer dimensions to fractal objects.

1925: Based on the concept developed by Julia in 1918, three mathematicians, Brauer, Hopft and Reidmeister wrote an essay in which a graphical representation of the Julia's work was presented for the first time in a conference in Berlin (Curtis, 1999). Nowadays, this graphical representation is known as the Julia's set, as shown in Figure 2.6.

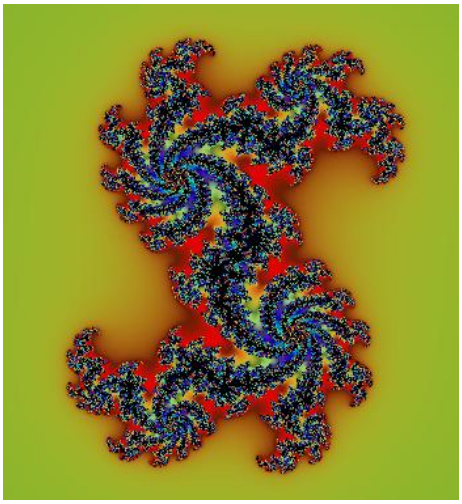


Figure 2.6: Visual representation of a Julia's set

1964: The term ‘self-similar’ was first utilized by Mandelbrot who was a research worker, in an internal IBM report (Peitgen and Saupe, 1988).

1968: Mandelbrot looked back to the problem of the coastlines of Britain described previously by Richardson (Mandelbrot, 1977). When Richardson saw only an empirical exponent α, Mandelbrot interpreted 1 + α as a dimension, and showed the fractal dimension of coastlines. He then proved that coastlines are part of a finite area while being of infinite length.

1975: Mandelbrot published a book called ‘Les Objets Fractals’ (Mandelbrot, 1977). This is the first book where the word ‘fractal’ appears to describe the entire different phenomenon previously mentioned. Nowadays, this book still remains as reference of the fractal geometry.

1979: Mandelbrot started to apply the concept of fractal to deterministic fields. This concept is important because most of the fractals that are known today come mainly from this family of fractals. It is at this time that Mandelbrot started studying the work of Julia (Falconer, 2003).

1980-1981: The first publication of ‘Mandelbrot Set’ as shown in Figure 2.7 was in 1980-1981. This picture is the most well known fractal (Peitgen and Saupe, 1988).

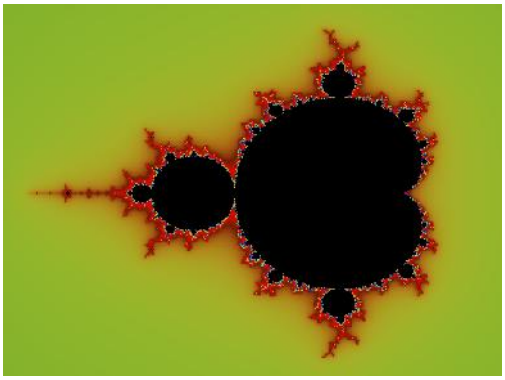


Figure 2.7: The Mandelbrot Set

1981: Diffusion-limited aggregation was introduced by Witten and Sander as a universal process. This process is independent of small changes in parameters and yields robust patterns which are relatively constant during growth (Vicsek, 1989).

2.2 Fractal Geometry

Fractal or fractional dimension is something that can never be understood inside the realm of elementary geometry. It is another field in which at least one of Euclid's postulates does not hold, and where other mathematical realities emerge. Thus, it can be said that there are two types of geometry: Euclidean and non-Euclidean geometries. The first group covers the plane geometry, solid geometry, trigonometry, descriptive geometry, projective geometry, analytical geometry and differential geometry. In the second group, there are hyperbolic geometry, elliptic geometry and fractal geometry. Almost all geometric forms used for building man-made objects belong to Euclidean geometry. They compromised of lines, planes, rectangular volumes, arcs, cylinders, spheres and defined shapes. These elements can be classified as belonging to an integer

dimension: 1, 2, or 3. Table 2.1 gives the summary of the major differences between fractal and the traditional Euclidean geometry.

Table 2.1: A comparison of Euclidean and fractal geometry (Peitgen & Saupe, 1988)

EUCLIDEAN	NON EUCLIDEAN (FRACTAL)
Traditional (>2000 yrs)	Modern monsters (~ 30 yrs)
Based on characteristic size or scale	No specific scaling
Suits man made objects	Appropriate for natural shapes
Described by formula	Recursive algorithm

Fractal geometry allows length measurements to change in a non-integer or fractional way when the unit of measurements changes. The governing exponent, *D* is called *fractal dimension* (Smith et al., 1990). The fractal dimension is a statistical quantity that gives an indication of how completely a fractal appears to fill space, as one zooms down to finer and finer scales. Fractal object has a property that more fine structure is revealed as the object is magnified, similarly like morphological complexity, which means that more fine structure (increased resolution and detail) is revealed with increasing magnification. Fractal dimension measures the rate of addition of structural detail with increasing magnification, scale or resolution. The fractal dimension, therefore, serves as a *quantifier of complexity*.

2.2.1 Self Similarity

The main idea behind fractal geometry is self-similarity. Self-similarity means that a structure (or process) can be decomposed into smaller copies of itself. This means that a self-similar structure is infinite. Self-similarity entails scaling. For an observable $A(x)$, which is a function of a variables x: $A = A(x)$, obeys a scaling relationship:

$$A(\lambda x) = \lambda^s A(x) \qquad (2.1)$$

where λ is a constant factor and s is the scaling exponent, which is independent of x. For example, in a three-dimensional Euclidean space, volume scales as the third power of linear length, whereas fractals scale according to their fractal dimension (Focardi, 2003). Approximate self-similarity means that the object does not display perfect copies of itself. For example a coastline is a self-similar object, a natural fractal, but it does not have perfect self-similarity. A map of a coastline consists of bays and headlands, but when magnified, the coastline is not identical but statistically the average proportions of bays and headlands remain the same no matter the scale (Judd, 2003).

It is not only natural fractals that display approximate self-similarity, the Mandelbrot set is another example. Identical pictures do not appear straight away, but when magnified, smaller examples will appear at all levels of magnification (Judd, 2003). *Statistical self-similarity* means that the degree of complexity repeats at different scales instead of geometric patterns. Many natural objects are statistically self-similar whereas artificial fractals are geometrically self-similar (Yadegari, 2001).

Geometrical similarity is a property of the space-time metric, whereas physical similarity is a property of the matter fields. The classical shapes of geometry do not have the matter field property; a circle, if on a large enough scale will look like a straight line. This is why people believed that the world was flat, the earth just looks that way to humans (Carr and Coley, 2003).

2.2.2 Fractal Dimension

Fractal dimension is a measure of how complicated a self-similar figure is. In a rough sense, it measures how many points lie in a given set. The fractal dimension is often fractional. However, in algebra, the dimension of a space is defined as the smallest number of vectors needed to span that space (Rucker, 1984). In the 3 dimensional space, mathematicians traditionally denote the coordinates of three orthonormal vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, but sets are usually not vector spaces. Nevertheless, for aggregates, a fractal dimensionality in terms of scaling relationship between two different aggregate's properties X and Y (e.g. mass and length) can be observed such as (Meakin, 1988):

$$Y \propto X\, d_f \tag{2.2}$$

where d_f is all-purpose fractal dimension as described by Meakin (1988).

Mandelbrot (1983) developed the 'concept of homothetic dimension' relative to geometric fractals. Let X be a complete metric space and let $A \subset$ X. If N (A, ϵ) is the least number of balls of radius less than ϵ that are needed to cover A, then the number $D(A)$ defined by

$$D(A) = \lim_{£ \to 0} \frac{\ln N(A, £)}{\ln \frac{1}{£}} \tag{2.3}$$

is called the fractal dimension of A.

For each part (N) of the fractal deducted from the whole and having a homothetic ratio $r(N)$, the fractal dimension d_f is defined as:

$$d_f = \frac{Log(N)}{Log\left(\frac{1}{r}\right)} \tag{2.4}$$

For example, for the Von Koch's snowflake iteration as illustrated in Figure 2.8, each side of unit 1 of a triangle is divided by 3, hence. r = 1/3. The central third of one side is replaced by 2 smaller lines of length 1/3. Therefore, one line is now subdivided into 4 smaller lines of length 1/3, hence N = 4. Its fractal dimension now becomes:

$$d_f = \frac{Log(4)}{Log(3)} \approx 1.26$$

Figure 2.8: Construction of the Von Koch's snowflake

2.2.3 Types of Fractals

Fractal geometry is the geometry of structures that have a scaling symmetry. The simplest types of fractals are self-similar fractals that are invariant to an isotropic change of length scale (Meakin, 1991). Another approach to fractals is the way they are

generated, for example by an *iterative* process. This process of iteration leads to different types of fractals. Generally fractals can be divided into two main types:

1. Deterministic Fractals
2. Random Fractals

In this thesis, much emphasis is given to the second type of fractals that is on the random type fractals.

2.2.3.1 Deterministic Fractals

Deterministic fractals are the first type of fractal generated by an iterative process. The term deterministic means that a simple process of iteration is applied to build the fractal such as the iteration of a complex function that generates the 'Mandelbrot Set' as shown in Figure 2.7. The iteration process is a geometrical transformation called *generator* on an object. This object is called *initiator*. For the construction of the so-called 'Koch's Curve' the transformation for each iteration is repeated. To build this fractal, a line of unit 1 is divided by 3 and the central $\frac{1}{3}$ is taken out and is replaced by 2 lines of length $\frac{1}{3}$. On the next iteration, the same transformation is applied on the remaining lines repeatedly. Its construction is described in Figure 2.9 as follows:

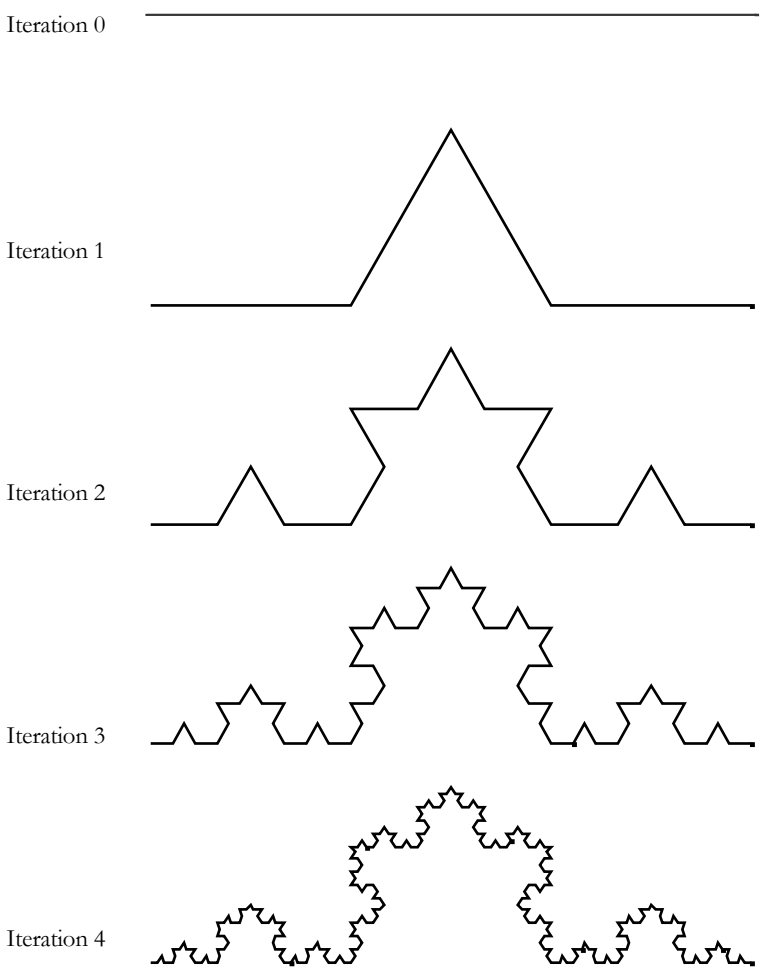


Figure 2.9: Construction of Koch's curve

An important property of this fractal is its length that is infinity. The length of the initiator is 1, therefore, after the first iteration; the calculated length of the object is 4 lines of length $\frac{1}{3}$, that is $\frac{4}{3}$. Then the second iteration gives 16 lines of length $\frac{1}{9}$. The length now becomes $\frac{16}{9}$. More generally, at each iteration n, the length is equal to $(4/3)^n$. As n tends to infinity, $(4/3)^{\infty}$ will approach ∞. The property of self-similarity can also be easily seen, as illustrated in Figure 2 10.

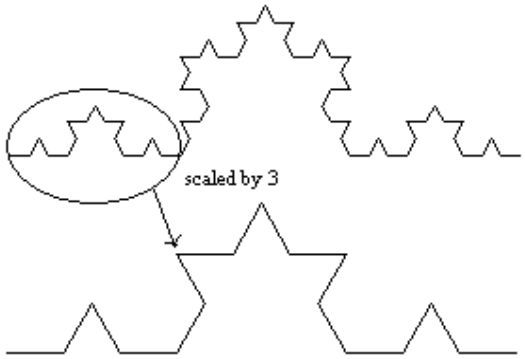


Figure 2 10: Self similarity property of the Koch Curve

2.2.3.2 Random Fractals

Random fractals are generated by stochastic processes, for example, trajectories of the Brownian motion, Lévy flight, fractal landscapes and the Brownian tree. The latter yields the so-called mass- or dendritic fractals, for example, diffusion-limited aggregation clusters. In the 1980's, Meakin developed different aggregation models in order to study the various ways an aggregate could be generated (Meakin, 1988; Meakin, 1991). Those aggregation models which are similar to the L-system are computer-generated where a set of transformation is applied on the generator that, in this case, would be an initial particle or cluster in the model. Random fractals have been used extensively in computer graphics to model natural objects (Ebert, 1996).

Many attractive images and life-like structures can be generated using models of physical processes from areas of chemistry and physics. One such example is diffusion limited aggregation (DLA) which describes, among other things, the diffusion and aggregation of zinc ions in an electrolytic solution onto electrodes. The term 'diffusion' is used due to the particles forming the structure wandering around randomly before attaching themselves

(aggregating) to the structure. The term ‘diffusion-limited’ is referred to because the particles are considered to be in low concentrations so they do not come in contact with each other and the structure grows one particle at a time rather than by chunks of particles. Other examples can be found in coral growth (Tanner, 1995), the path taken by lightning (Gulyás and Szedenik, 2009), coalescing of dust or smoke particles (Lehtinen et al., 1996), and the growth of some crystals (Zhi-Qiang Zou, 2011).

2.3 Significance of Fractals

The term fractal has always been associated with the complex geometric shapes which can be characterized by non-integer dimensions. Generally, fractals can be found in unbalanced phenomenon either naturally or experimentally developed in laboratories. The fractal concept has been used in many fields like chemistry, biology, medicine, weather forecast and engineering where it provides understanding of the extraordinary patterns and chaos (Radnoczy, 1987, Niemeyer, 1984, Chandra, 1996).

2.3.1 Fractals in Physical Sciences

Fractals obviously generate some convincing models of natural phenomena such as mountains and clouds for use in computer graphics imagery, and they provide very compelling abstract pictures. But in the past two decades, about one third of all physics papers submitted to journals for publication at least mentioned fractals somewhere (Musgrave, 1993). It is also known that many universities all around the world have now

offered courses on the subject fractals mainly concerning the field of mathematics and physics.

Looking at fractals in mathematics, some fractal patterns exist only in mathematical theory, but others provide useful models for the irregular yet patterned shapes found in nature such as the branching of rivers and trees. Mathematicians tend to rank fractal dimensions on a series of scales between 0 and 3. One-dimensional fractals (such as a segmented line) typically rank between 0.1 and 0.9, two-dimensional fractals (such as a shadow thrown by a cloud) between 1.1 and 1.9, and three-dimensional fractals (such as a mountain) between 2.1 and 2.9. Most natural objects, when analyzed in two dimensions, rank between 1.2 and 1.6 (Ouellette, 2001).

The nonlinear mathematics models nature more accurately, but is intractable in comparison to the linear approximations. When computers made it possible for scientists to begin to cope with these previously-intractable nonlinear systems, they discovered something very surprising which is in any perturbation to the initial state of the system, no matter how small or seemingly insignificant, will cause the system to diverge; that is to evolve into an arbitrarily different future state, within a finite period of time. This discovery is known as *deterministic chaos* or *sensitivity to initial conditions*.

2.3.2 Fractals in Biological Sciences

Biologists have traditionally modeled nature using Euclidean representations of natural objects or series. Examples include the representation of heart rates as sine waves, conifer trees as cones, animal habitats as simple areas, and cell membranes as curves or

simple surfaces. However, scientists have come to recognize that many natural constructs are better characterized using fractal geometry. Biological systems and processes are typically characterized by many levels of substructure, with the same general pattern being repeated in an ever-decreasing cascade. Relationships that depend on scale have profound implications in human physiology (West and Goldberger, 1987), ecology (Loehle, 1983, Wiens, 1989), and many other sub-disciplines of biology. The importance of fractal scaling has been recognized at virtually every level of biological organization.

Fractal geometry proves to be a unifying theme in biology (Kenkel and Walker., 1993) since it permits generalization of the fundamental concepts of dimension and length measurement. Most biological processes and structures are non-Euclidean, displaying discontinuities, jaggedness and fragmentation. Classical measurement and scaling methods such as Euclidean geometry, calculus and the Fourier transform assume continuity and smoothness. However, it is important to recognize that while Euclidean geometry is not fully realized in nature, neither is strict mathematical fractal geometry. Specifically, there is a lower limit to self-similarity in most biological systems, and nature adds an element of randomness to its fractal structures. Nonetheless, fractal geometry is far closer to nature than is Euclidean geometry (Deering and West, 1992).

The relevance of fractal theory to biological problems is dependent on objectives. To the forester interested in estimating stand board-feet, a Euclidean representation of a tree trunk (as a cylinder or elongated cone) may be quite adequate. However, for an ecologist interested in modeling habitat availability on tree trunks (say, for small epiphytes or invertebrates), fractal geometry is more appropriate. Using the approach of fractal geometry, the complex surface of tree bark is readily quantified.

A forester's diameter tape ignores the surface roughness of the bark, giving but a crude estimate of the circumference of the trunk. For an insect 10 mm in length, the distance that it must travel to circumnavigate the trunk is much greater than the measured circumference value. For an insect of length 1 mm, the distance traveled is even greater. This has consequences on the way that the tree trunk is perceived by organisms of different sizes. If the bark has a fractal dimension of D = 1.4, an insect an order of magnitude smaller than another perceives a length increase of $10^{D-1} = 10^{0.4} = 2.51$, or a habitat surface area increase of $2.51^2 = 6.31$. By contrast, for a smooth Euclidean surface, D = 1 and both insects perceive the same 'amount' of habitat. The higher the fractal dimension D, the greater the perceived rate of increase in length (or surface) with decreasing scale.

2.4 Fractal Growth Models

Many fractal growth phenomena found in experiments and numerical simulations explored the properties of aggregation kinetics, gelation, and sedimentation (Aharony, 1991). The aggregation of particles often produces fractal clusters. A typical aggregate is the commonly known computer generated simulation of ‘diffusion limited aggregation’. The shape looks very similar to those that arise in many natural aggregation processes, including diffusion limited electrodeposition (Matsushita et al., 1984a), growth in aqueous solutions (Sawada et al., 1986), dielectric breakdown (Niemeyer et al., 1984), viscous fingers in porous media (Maloy et al., 1985), and fungi and bacterial growth (Matsuura and Miyazima, 1992, Matsuyama et al., 1993, Ben-Jacob et al., 1994).

To describe these aggregates, one must first characterize their structures quantitatively (Aharony, 1991). Characterization on its fractal dimensionality, or exponents, each of which determines one of its physical properties is very important. Growth models are used to understand the relationship between the microscopic interactions which are responsible for its growth, and the specific complex macroscopic shapes. This is done by setting up a few simple microscopic growth rules, by which particles are added to the aggregate and with repeated iteration it gives rise to the macroscopic cluster. Some of the well known fractal growth models normally used for simulation of fractals are described in the following sections.

2.4.1 Eden Model

The Eden model is the simplest growth model (Eden, 1961) and the one that probably applies in most cases. Starting from an initial seed, a new particle is added to cluster on one of the surface sites. A surface site here is defined as a site sharing a side with the existing cluster. The way in which the surface site is chosen can vary. One version of the Eden model selects with equal probability among all the surface sites where a new particle will be added. Another version counts the number of neighbors of each surface site and the probability that a new particle is added is directly proportional to the number of neighbors. The third version of the Eden model chooses a ’mother cell’ with equal probability among the particles which are not completely surrounded by other particles.

2.4.2 Percolation Model

The randomness of a fluid spreading through a medium maybe of two quite different types (Feder, 1988). The first type is the random walks of the fluid particles in the familiar diffusion processes. The other case in which the randomness is frozen into the medium itself and it is known as a 'percolation process', since it behaves like coffee in a percolator (Broadbent and Hammersley, 1957).

Compared to diffusion process where a diffusing particle may reach any position in the medium, percolation process has a feature, where there exists a 'percolation threshold', under which the spreading process is confined to a 'finite' region. For example, spreads of blight from one tree to the other in an orchard where the trees are planted on the intersections of a square lattice. Here, when the spacing between the trees is increased so that the probability for infecting a neighboring tree falls below a critical value, p_c, then the blight will not spread over the orchard. Thus, the value of the percolation threshold has to be determined by simulations.

2.4.3 Ballistic Deposition Model

Ballistic deposition was introduced as a model of colloidal aggregates, and early studies concentrated on the properties of the porous aggregate produced by the model (Family, 1990, Horvath et al., 1991). The particles in the ballistic deposition model follow a straight-line trajectory until they first encounter a particle on the surface, or a particle in one of the nearest-neighbor columns. As soon as a particle reaches such a position, it permanently sticks to the surface and becomes part of the deposit. Evolution of an

interface in a ballistic deposition model can be described by the dynamic scaling approach (Family and Vicsek, 1985). Moreover the surface of the deposit is a self-affine fractal, since the atoms are not allowed to diffuse on the surface.

2.4.4 Dielectric Breakdown Model

Dielectric breakdown refers to the formation of electrically conducting regions in an insulating material exposed to a strong electric field. For example, the intense electric fields during thunderstorms can produce a conducting path in the air along which many electrons flow (lightning). A formal model, ignoring the physical details of the processes, was proposed in 1984 by Niemeyer, Pietronero and Weismann (Niemeyer et al., 1984). Dielectric breakdown patterns exhibit a branching, fractal pattern with a dimension of about 1.7.

2.4.5 Viscous Fingering Model

In viscous fingering, the principal force is due to viscous forces in the defending fluid (Aharony, 1991). The process is obtained by injecting a low viscosity fluid into a medium of high viscosity fluid with a high injection rate. The capillary effects and the pressure drop in the invading fluid are negligible. The structures typically consist of fingers of invading fluid that propagate through the medium with only a few small trapped clusters of defending fluid left behind.

Viscous fingering was first studied in a Hele-Shaw channel where one observes fingering patterns when glycerol is displaced by air (Saffman and Taylor, 1958). A

Hele-Shaw cell consists of two transparent plates separated by a given distance and the patterns obtained are fully described by Darcy's equation and the capillary pressure due to the interfaces between the two phases. In 1985 Chen and Wilkinson (Chen and Wilkinson, 1985) and Maloy and coworkers (Maloy et al., 1985) studied viscous fingering in a porous medium where they concluded that the disorder of the system has significant effect on the fingering process.

2.4.6 Diffusion Limited Aggregation Model

Diffusion limited aggregation (DLA) is a model of irreversible growth to generate fractal structures as proposed by Witten and Sander (1981). It has been used to study a great variety of processes including dendritic growth, viscous fingering in fluids, dielectric breakdown and electrochemical deposition. The model is set by the following simple rules:

> A seed is fixed at the origin of some coordinate system and one particle is released from a far-away boundary and allowed to take random walks (diffuse). If the particle touches the seed, it irreversibly sticks to the seed and forms a two-particle aggregate. As soon as the random walker is removed either by being captured or escaping the boundary, the next walker is released and the process is repeated. Now it can stick to any particle in the aggregate as well as the original seed.

The resulting clusters are highly branched since DLA enhances the instability of growth. The arriving particles are far more likely to stick to the tips of outer branches than to maneuver their way deep into the *fjords* (narrow inlet of a section) before contacting the

surrounding branches. Thus the tall branches of the cluster screen the small ones and grow faster. The growth on the tips, however, is not always in the outward radial direction. Sometimes a few new branches are spun off from one tip site as occurred in the original seed. The *tip-splitting* makes the DLA clusters a self similar fractal.

2.5 Fractal Growth Patterns Identification and Simulation

It has been identified that the formation of fractals without using any external stimuli resulted into isotropic DLA patterns as reported by Chandra (1996). Since the fractals studied in the present work were obtained without applying any external stimuli, DLA model which is based on the Brownian motion theory was thought to be suitable simulation technique. The summary of the simulation technique is given in the following subsection.

2.5.1 Simulation of Fractals Using DLA Model

DLA is one of the most important models of fractal growth. It refers to a simple growth algorithm in which individual particles are added to a growing cluster through diffusion-like process. Starting from any suitable immobile aggregate seed in a plane, a new particle is launched at a random position far away from the aggregate seed and is allowed to undergo Brownian motion. When the random walking particle touches the seed, it is stopped and incorporated to the aggregate. The process of launching a random walker and adding it to the aggregate on its first contact is repeated until the aggregate

reaches a desired number of particles (Witten and Sander, 1981). Figure 2.11 gives a visual representation of the above mentioned process.

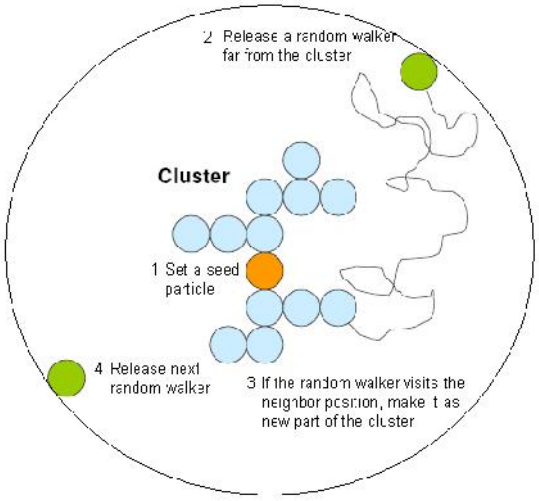


Figure 2.11: A model representing aggregation of cluster particles

2.6 Methods for Determination of Fractal Dimension

Fractal dimension is a statistical quantity that gives an indication of how completely a fractal appears to fill space, as one zooms down to finer and finer scales. There are many specific definitions of fractal dimension. Summary of some of the more commonly used methods for determination of fractal dimension of natural forms are presented in this section. These methods include the information dimension method, mass dimension method and box counting method.

(i) Information dimension method

This method requires the use of boxes but is generally different fom the box counting method. It does not enter into calculation the number of boxes occupied regardless of whether it contains one point of relatively large number of points. Instead, the information dimension effectively assigns weights to the boxes in such a way that boxes containing a greater number of points count more than boxes with fewer points (Dierking, 2001). The fractal dimension D_i is given from the proportionality

$$I(d) \sim -D_i \log(d) \tag{2.5}$$

where $I(d)$ is the information entropy of $N(d)$ boxes of size d, given by

$$I(d) = -\sum_{i=1}^{N(d)} m_i \log(m_i) \tag{2.6}$$

with $m_i = \frac{M_i}{M}$ where M_i is the number of points in the ith box and M the number of total points in the data set.

(ii) Mass dimension method

The mass dimension method also known as the Scholl method, which yields the fractal dimension D_m, following the proportionality

$$m(r) \sim r^{Dm} \tag{2.7}$$

where $m(r) = M(r)/M$ is the 'mass' within a circle of radius r, where $M(r)$ is the data set of points contained within a circle and M the total number of points in the set. If the set is a fractal, the plot of log $m(r)$ versus log r will follow a straight line with a positive slope equal to D_m (Dierking, 2001). This method is best suited to objects that follow some radial symmetry, such as the dendritic growth in radial axis.

(iii) Box-counting dimension:

In fractal geometry, the box-counting dimension is a way of determining the fractal dimension of a set S in a Euclidean space R^n. To calculate this dimension for a fractal S, imagine this fractal lying on an evenly-spaced grid, and count how many boxes are required to cover the set. The box-counting dimension is calculated by seeing how this number changes as the grid becomes finer.

Suppose that $N(s)$ is the number of boxes of side length s required to cover the set (Hastings and Sugihara, 1993). Then S has box dimension D if $N(s)$ satisfies the power law

$$N(s) \approx c(1/s)^{D} \tag{2.8}$$

asymptotically in the sense that

$$\lim_{s \to 0} N(s) s^{D} = c \tag{2.9}$$

By solving equation (2.8) asymptotically for D, the box-counting dimension is computed as:

$$D = \lim_{s \to 0} \left[-\frac{\log N(s)}{\log s} \right] \tag{2.10}$$

This method is a favorite among most researchers and is considered the easiest to perform (McNamee, 1991). The box-counting dimension can be used to analyze irregularities in surfaces filling space volume and suitable for images, however complex. The use of a mesh grid overlapped over a structure allows the box counting method to conduct both textural and structural analysis of a structure. In addition, the mesh grid also allows the analysis of objects scattered in an image and this method can be adapted to measure objects or processes in multiple dimensions (Cross, 1997).

C H A P T E R 3

RESEARCH METHODOLOGY

This work covers both experimental and simulation work. The research work is divided into 4 parts.

1. The first part involves experimental work in a laboratory for obtaining fractals. In the experimental work, a number of polymer membranes prepared using different types of host polymers and inorganic salts were employed as the media for fractal growth.

2. In the second part of this work, the simulation of the fractal growth patterns was carried out. Suitable fractal growth model was identified and utilized for simulation to reflect the fractal growth patterns obtained in the laboratory.

3. In the third component, development of computer program for simulation design was done. The simulation program was then validated by running the simulation for other similar types of fractal growth patterns such as electrochemical deposition, Hele-Shaw fingering and bacterial growth. For this purpose, the fractal dimensions of the simulated patterns were acquired and corroborated.

4. In the final stage, simulation of single and multiple clusters of the fractal growth patterns observed in polymer membranes was done. The characteristics of the simulated fractal patterns were compared with those of the experimentally obtained fractals.

3.1 Experimental Methods

3.1.1 Preparation of Polymer Membranes as Media for Culturing Fractals

Chitosan, PEO and PVDF-HFP/ PEMA polymer electrolyte membranes, were prepared using solution casting method. Chitosan-silver nitrate ($AgNO_3$) membranes were prepared by dissolving chitosan and $AgNO_3$ with 1% acetic acid solutions. PEO (mol wt. ~6×10^5) and ammonium iodide (NH_4I) were weighed in desired PEO:NH_4I weight ratios and dissolved in 100 ml acetone. PVDF-HFP, PEMA and ammonium trifluoromethanesulfonate ($NH_4CF_3SO_3$) of fixed w% ratio were dissolved in N-N Dimethyl formamide (DMF) at 40 ℃ using digital magnetic stirrer. Appropriate weight percent of chromium oxide (Cr_2O_3) in nanosize (particle size: 35-85 nm) was added to the polymer-salt solutions of PVDF-HFP/PEMA-$NH_4CF_3SO_3$. In this preparation, all the mixtures were stirred for about 10-12 hours until homogeneous solutions were obtained. The solutions were then cast into Petri dishes and allowed to evaporate slowly at room temperature to form films and then kept in desiccators containing silica gel for further drying. After a few months, fractals were found to grow in these films. After drying, complex geometric patterns were observed. The processes for the preparation of the media to grow fractals are depicted in Figure 3.1(a)-(f).

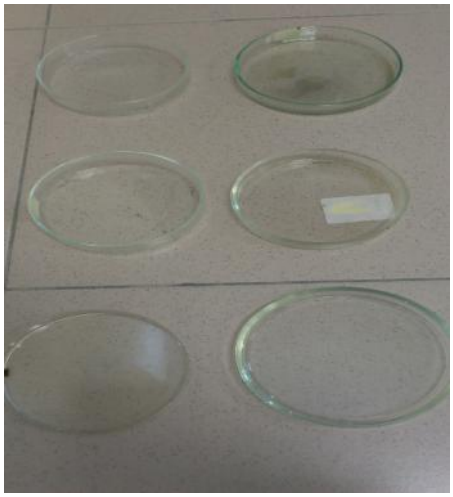
(a) Identifying of suitable chemicals

(b) Weighing of chemicals

(c) Mixing of chemicals

(d) Stirring of mixtures

(e) Casting of solution into Petri dish



(f) Placing samples in dry and dark place

Figure 3.1: The processes of preparing polymer membranes

3.2 Simulation Models

As described earlier in section 1.1, there are three basic models of fractal growth: percolation (Bunde and Havlin , 1991), particle-cluster aggregation (PCA) (Vicsek, 1992; Tan et al., 1999; Meakin, 1983) and cluster-cluster aggregation (CCA) (Tan et al., 2000; Zhang and Liu, 1998). Among the models of PCA, DLA (Witten and Sander, 1983) is the most well-known. DLA is a phenomenon that is observed in many areas of nature, ranging from physics to geology to chemistry to biology to meteorology. Random fractal growth processes essential condition can be described by DLA model. The fractals obtained in this study bear close resemblance to patterns of similar DLA type fractals such as found in dendritic growth and electrochemical deposition. As such DLA model was considered to be a worthy simulation model for this study. The model is set by the following simple rules:

A seed is fixed at the origin of some coordinate system and one particle is released from a far-away boundary and allowed to take random walks (diffuse). If the particle touches the seed, it irreversibly sticks to the seed and forms a two-particle aggregate. As soon as the random walker is removed either by being captured or escaping the boundary, the next walker is released and the process is repeated. This new walker can stick to any particle in the aggregate as well as seed particle.

The resulting clusters are highly branched since DLA enhances the instability of growth. The arriving particles are far more likely to stick to the tips of outer branches than to maneuver their way deep into the fjords (narrow inlet of a section) before contacting the surrounding branches. Thus the tall branches of the cluster screen the small ones and grow faster. The growth on the tips, however, is not always in the outward radial direction.

Sometimes a few new branches are spun off from one tip site as occurred in the original seed. The tip-splitting makes the DLA clusters a self-similar fractal.

DLA-clusters are aggregates, where the shape of the cluster is in a way controlled by the possibility of particles to reach the cluster. The aggregates may grow as long there are particles moving around. During the growth it may happen, that arms of the cluster catch particles so that they cannot reach inner parts of the cluster. Older (inner) parts of the cluster cannot catch new particles as the younger, outer parts of the cluster come into their way. During the diffusion of a particle through a medium it is more likely, that it attaches to the outer regions than to the inner ones of the cluster. Thus, a fluffy shape occurs, with many arms, like corals or trees. The volume is not filled in entirety, but there are many gaps.

As there are lots of models in the field of investigation of fractals, computation is a way to compare the nature with the models. For DLA-cluster one approach is to simulate the random walk of the particles and their aggregation. Typically one uses a lattice, puts an initial seed particle at some origin and another particle somewhere on the lattice. Then the second particle moves around in random motion, step by step from lattice site to lattice site. Finally it will meet the first particle. Then another particle is thrown onto the lattice, it walks around and after a while meets the first two. This is continued for as many particles as one likes, one after the other (Witten and Sander-model)(Witten and Sander, 1983).

As a result, the whole process would take up much time that would slow simulation, since most of the time the particles move around and around and only rarely come into the vicinity of the cluster, where they might stick. This problem becomes much more

relevant, when the cluster grows and the amount of gaps between the arms is by far bigger than the amount of lattice sites in the vicinity of the cluster. Thus, many computations of big clusters had to be done on fast mainframe machines. But various approaches (Vicsek, 1992) came up to reduce this time while maintaining the random character and the diffusion control of the mechanism.

In this work, the simulation model implements some variations:

i. Introduction of a sticking coefficient, so that not always when a particle reaches the cluster it will stick. Thus, when it does not stick immediately, it moves along in the vicinity of the cluster's arms, until it either finally sticks somewhere or gets lost. A further modification can be done by using different attaching probabilities depending on the geometrical environment, e.g. the more neighbors are already present, the more likely it is for a particle to attach. Either way the outcome could be of different characters in shape (from needle like to cauliflower like).

ii. Varying the lattice geometry by using a square lattice with four or eight neighbor sites or even a triangular lattice with six neighbors. Typically, an overall shape of the cluster is related to the shape of the lattice.

iii. Extending the model to several simultaneously growing cluster within a "solution" of free jittering particles whereby the cluster-cluster-aggregation (CCA) could be employed, since when cluster grow and grow, at some point they may meet and attach to each other.

In this work, the computer programs developed were written in MATLAB as a computer coding programming platform.

3.3 Development of Simulation Programs

The simulation work done in this study involved devising computer programming of two separate fractal conditions. One is for simulation of single cluster fractal and the other is for multiple clusters. In both programs, important aspects of fractal formation parameters were implemented according to the suitability of the fractal patterns being simulated. Computer simulations were performed on the platform of MATLAB Version 7.12 (R2011a). MATLAB was chosen as it allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. It can solve technical computing problems faster than with traditional programming languages. Apart from that, MATLAB also provides MATLAB Compiler Runtime (MCR) used for simplifying distribution of compiled applications and components.

3.3.1 Development of Simulation Program for Single Cluster Fractal Pattern

For simulation of single cluster fractal growth pattern, the traditional DLA model (Witten and Sander, 1981) was applied. DLA is modeled by tracking random walking particles and a cluster to which the random walking particles may stick. The cluster is composed of a seed, the initial particle member, and other particles that have stuck to the cluster. The random walkers are released one by one, only one on the lattice at a time, to wander around until they land in a position touching a cluster member as shown in Figure 3.2. If it touches, it sticks. Then the next particle is released. The details of particle release and sticking are dependent on the practical limitations of the simulation. The particles cannot be released too far from the cluster or it will take too long for the particle to find

the cluster. The particle cannot be released too close to the cluster, since the statistical profile of the diffusion must have correlation lengths much longer than the size of the cluster itself (Hac et al., 2005). If the particle moves too far away, the particle should be killed since it would take too long for the particle to happen to wander back to the vicinity of the cluster. The particle should not be killed too close to the cluster since the effective diffusion lengths must be much longer than the size of the cluster itself (Li et al., 2009).

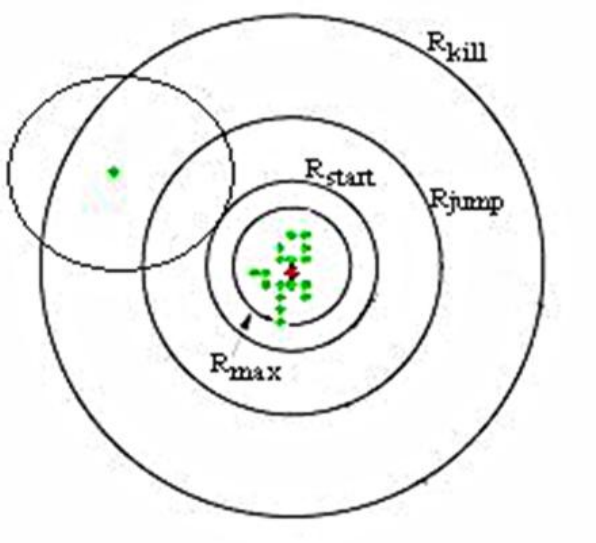


Figure 3.2: An off the scale model of aggregation of cluster particles (Biehl, 2005)

The diffusive process satisfies Laplace's equation in the vicinity of the cluster. Since the cluster is considered a perfect absorber, the flux of particles on the surface is zero, and the surface is equivalent of an equipotential. From this perspective, the impact of a radius that kills random walking particles is also one of being an equipotential surface. The impact of the outer radius was treated by the method of images. The image of the cluster in the killing radius is that of a mirror cluster a long way away from the cluster and completely enclosing it. Any features in the cluster are mirrored in the image.

Based on the DLA model (Witten and Sander, 1981), the basic algorithm (Biehl, 2005) used in this work that employs two-dimensional lattice is as follows :

(1) INITIALIZE

Start with an immobile seed particle in the center of an otherwise empty lattice (cluster mass $M = 1$, cluster radius $R_{max} = 1$).

(2) LAUNCH A NEW PARTICLE

Place a single particle on a randomly chosen site with distance $R_{start} > R_{max}$ from the origin.

(3) DIFFUSION

Calculate the distance r of the particle from the origin.

If $r < R_{jump}$: Move the particle from its current position to a randomly chosen nearest neighbor site.

If $R_{kill} > r \geq R_{jump}$: Move the particle with equal probability anywhere on a circle with radius $(r - R_{start})$ around its current position.

If $r \geq R_{kill}$: Remove the particle from the lattice, go to step (2).

Repeat (3) until a nearest neighbor of a cluster site is reached, and then go to (4).

(4) AGGREGATION

Add the particle to the cluster, increase M by one and evaluate R_{max}. Stop if the desired mass M is reached, else go to step (2).

A particle which approaches the cluster from very far would enter a circle with Radius $R > R_{max}$ (for the first time) with equal probability anywhere on its circumference. This fact

is reflected in step (2), i.e. the launching of new particles in the above defined algorithm. Under the constraint that the circle must not interfere with the cluster, R_{start} was chosen to be as small as possible (e.g. $R_{start} = R_{max} + 2$). As the particles can only have integer coordinates, measured in lattice constants, the start position needs to be approximated by the closest lattice site.

A particle can drift arbitrarily far away from the origin, even though it was started very close to the cluster. Hence, a straightforward simulation would use a large portion of its computing time on the diffusion of particles far away from potential growth sites. In order to overcome this difficulty, another radius, $R_{kill} > R_{max}$ was introduced. Simply remove a particle if its distance from the origin exceeds R_{kill} and replace it with a new one randomly at R_{start}. The weakness of this simple idea of *killing* of particles is that a particle would eventually re-enter the start circle at a position which is correlated with its location at R_{kill}. Only in the limit $R_{kill} \rightarrow \infty$ these correlations vanish, so this parameter should be set to a rather large value.

Even if the diffusion is limited to within a distance R_{kill} as outlined above, large portions of the simulation will move the particle back and forth in empty regions with no potential growth sites. This time consuming Brownian motion far away from the cluster can be replaced by appropriate *shortcuts* using an additional parameter R_{jump} which satisfies $R_{start} < R_{jump} < R_{kill}$ as shown in (3).

Figure 3.3 gives an illustration of the implementation of the above mentioned DLA algorithm.

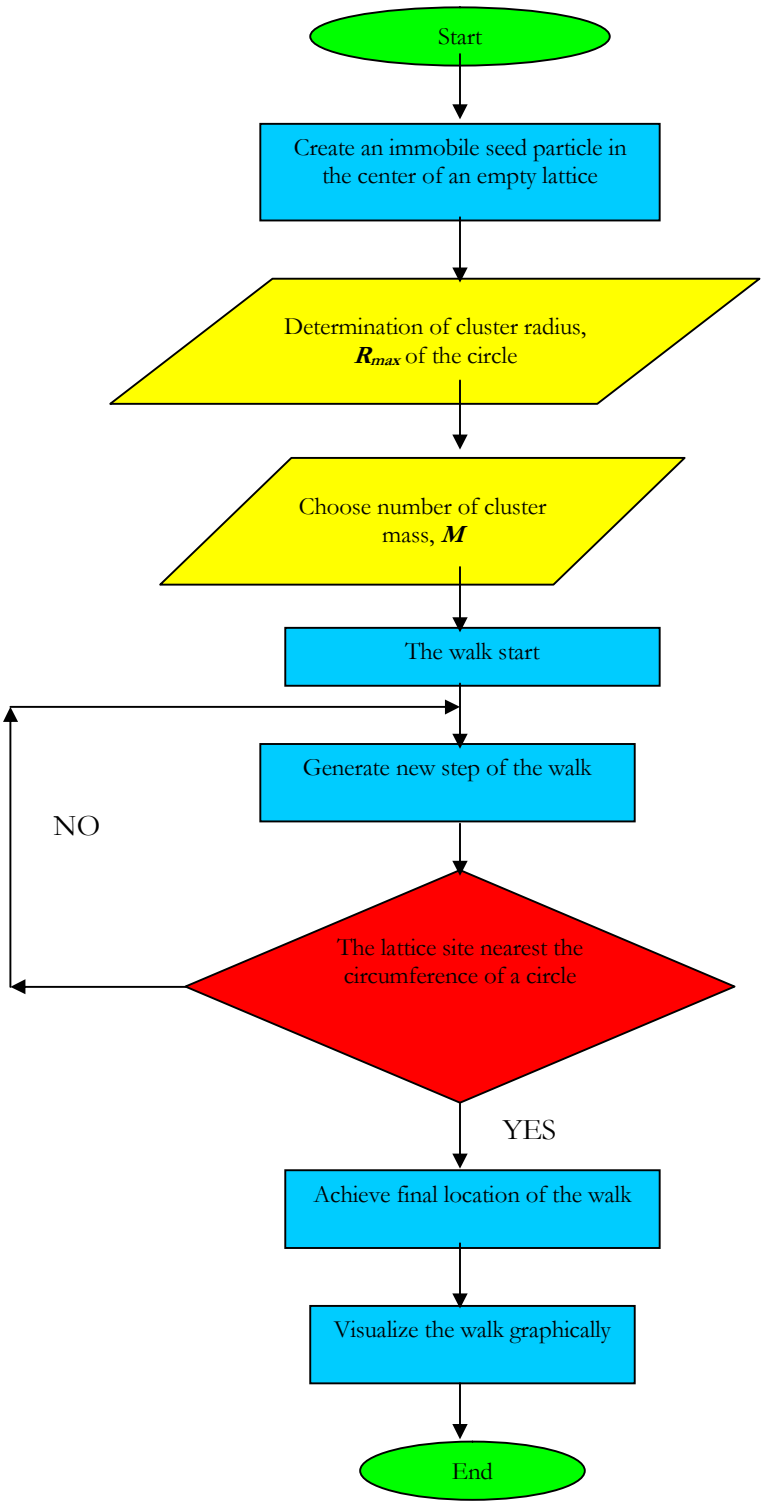


Figure 3.3: An illustration of the implementation of the algorithm for simulation of DLA model in 2D lattice

A program was then developed using these general code fragments assembled in MATLAB Version 7.12 as the computer language base. The following steps give the implementation of the mentioned algorithm.

Step 1. Set lattice size (latt) and center site position (cp) as

```
latt = 1001;
cp   = floor(latt/2);
```

with a region /area of a lattice site and the center element occupied = 1

```
coor = zeros(latt,latt);
coor(cp,cp) = 1;
```

Step 2. Launching of the particles one by one is given by

```
for ipart = 2: maxmass
```

setting the radius with the conditions

```
rstart = rmax + r;
rjump  = rstart + 10;
rkill  = 10 * rmax + 10;
```

note: for the medium size clusters it suffices to use rkill = 10 * rmax + 10;

Step 3. Diffusion of particles takes place with implementation of *shortcuts* with three main conditions:

Condition 1: $r \geq R_{kill}$

```
if distance > rkill
pos = circle(rstart,cp,cp);
ix  = pos(1);
iy  = pos(2);
distance = sqrt((ix-cp)^2 + (iy-cp)^2);
```

Condition 2: $r < R_{jump}$

```
if distance < rjump
check_1 = 0;     % Current position is empty.
oldx    = ix;
oldy    = iy;
```

```
end
```

Condition 3: $R_{kill} > r \geq R_{jump}$

```
else
      pos = circle((distance-rstart),ix,iy);
      ix = pos(1);
      iy = pos(2);
end
```

where *r* now is given by

```
distance = sqrt((ix-cp)^2 + (iy-cp)^2);
```

Step 4. Aggregation comes to effect when next neighbor (NN) site of cluster is reached

```
if distance < rstart
```

then set the coordinate as occupied site = 1 as given by

```
coor(ix,iy) = 1;
```

Finally, once the particle sticks, determine mass and maximum radius:

```
m = m+1;
rmax = max(rmax,sqrt((ix-cp)^2 + (iy-cp)^2));
```

In this work, the simulation runs a DLA model with single seed on 4, 6, or 8 lattice sites. There is only one output which is a list of particle coordinates called 'singleCluster'. There are four inputs needed; radius (r), maximum number of particles (maxmass), sticking coefficient (alpha), and type of lattice site (nlattice).

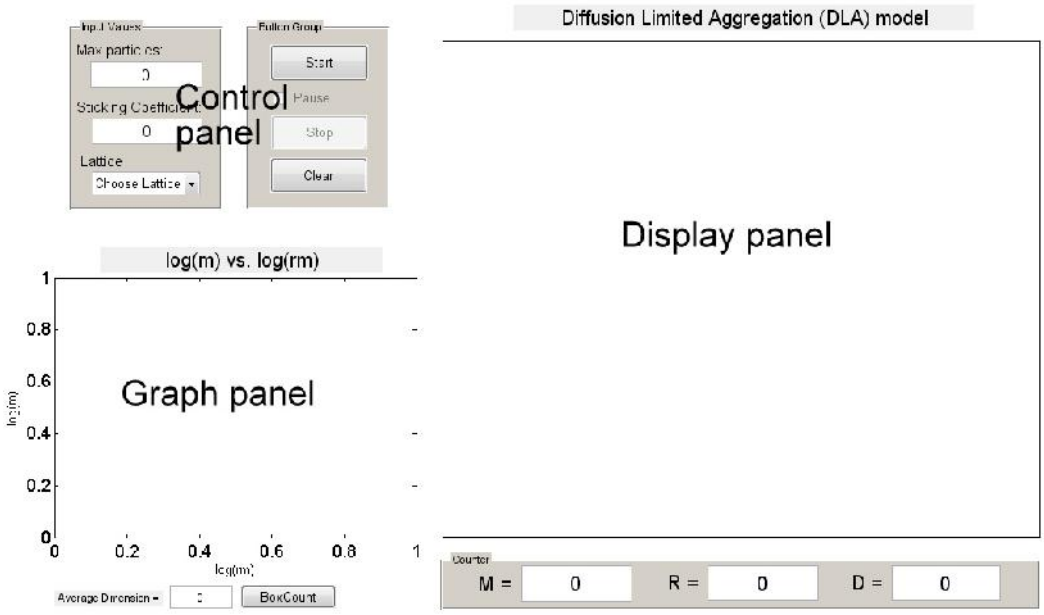


Figure 3.4: The main window of single seed DLA model program

As shown in Figure 3.4, the main window of single seed DLA model program developed in this research work consists of three panels: the control panel on the top left, the graph panel on the bottom left and the display panel on the right.

The control panel contains all the buttons and drop down menu to assign the values of the parameters for the simulation. The graph panel is used to draw the log-log plot of the calculation of fractal dimension. While the 2D image of the object can be shown on the display panel.

In order to run the simulation of a single cluster DLA fractal pattern, firstly the input values: “Max particles”, “Sticking coefficient” and “lattice” are set according to the desired values. For “Max particles”, only positive number values can be input while for the “Sticking coefficient” values of between and including 0 and 1 should be inserted. The drop down menu allows the selection for “lattice” of 4, 6 or 8 lattice sites for any one time

a simulation is being run. After setting the values for all these parameters, simulation image is visualized by the display panel by clicking the “start” button. This is accompanied by the pause button to temporarily stop the running simulation and also the stop button that can be clicked to stop the simulation.

Once the running simulation is complete, the values of Max particle, *M*, final size (radius), *R* and fractal dimension, *D* are given in the “counter” section below the display panel. Finally, the graph panel will show the log-log plot for the calculation of fractal dimension. Here, the fractal dimension of the simulated image is calculated from the slope of the linear regression curve of the log-log plot and the value is shown in the box below the graph.

There are two function files written in MATLAB specifically for the simulation program of the single cluster fractal pattern, they are the main function file, myDLAmodel.m and the Graphical User Interface (GUI) program file, myDLAdisplay.m file. Details of the programming files are shown in Appendix A and B.

3.3.2 Development of Simulation Program for Multiple Clusters Fractals

As a model approximation, the growth of multiple clusters can be considered as the addition of particles one at a time to sites of a two-dimensional squared grid. Then, according to DLA approach, the particle motion is described as a random walking over the grid. The added particle moves until it contacts the growing aggregate.

The number of such aggregates is determined by number of initial growth sites. In this work, the DLA model takes into account multiple seeds on 3 different lattice sites. There are four inputs needed to run this simulation; maximum number of particles (maxmass), sticking coefficient (alpha), number of seeds (nmax) and type of lattice site (nlattice).

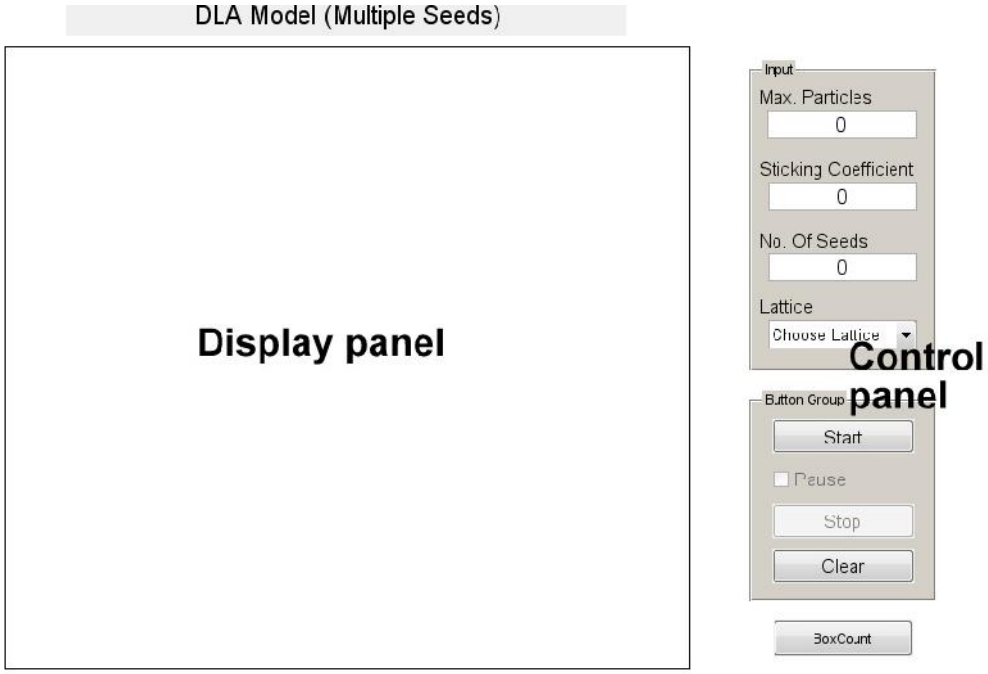


Figure 3.5: The main window of multiple seeds DLA model program

Figure 3.5 shows the main window of multiple seed DLA model program. The program developed in this research work consists of two panels: the control panel on the right and the display panel on the left.

The control panel contains all the buttons and drop down menu to assign the values of the parameters for the simulation. The display panel gives the 2D image of the simulated pattern.

For the simulation of multicluster DLA fractal pattern, firstly the input values: “Max particles”, “Sticking coefficient” and “No of seeds” are set according to the desired values. Similar to the single cluster simulation program, for “Max particles”, only positive number values can be input while for the “Sticking coefficients” values of between and including 0 and 1 should be inserted. An additional counter in this multicluster simulation program is the “No of seeds” box counter. One can choose the number of seed(s) by using this “No of seeds” counter. The drop down menu allows the selection for “Lattice” of 4, 6 or 8 lattice sites for any one time a simulation is being run. By clicking the “start” button, simulation image is visualized on the display panel. Together with the pause button to temporarily stop the running simulation is also the stop button that can be clicked to stop the simulation.

Generally, for the simulation program of the multiple clusters fractal patterns there are two function files written in MATLAB. They are the main function file, myDLAMultiple.m and the GUI program file, myDLAmodified.m file. Details of the programming files are shown in Appendix C.

3.4 Fractal Dimension and Fractal Growth Analysis

Fractal dimension is a statistical quantity that gives an indication of how completely a fractal appears to fill space, as one zooms down to finer and finer scales. There are many specific methods for determining fractal dimension. These methods include the information dimension method, mass dimension method and box counting method. Moreover, the physical meaning of fractal dimensions calculated by different methods may be different. To calculate box-counting fractal dimension, D_b, grids with

side length l, were laid over a threshold binary image of a fractal growth pattern. The grids containing more than a threshold value is defined as the occupied grids. The number of occupied grids N_i is counted for a series of grid side length l_i. Considering the fractal growth pattern as a fractal object, the number of occupied grids has a power law relationship with the grid side length, so the plot of the $ln(N)$ with $ln(l)$ is linear. The negative of the slope of this log-log plot is the box-counting fractal dimension, D_b, as such

$$D_b = -\frac{\Delta \ln N}{\Delta \ln l} \tag{3.1}$$

The concept of particle-counting fractal dimension, D_f, is derived from the mass fractal dimension, D_m, where D_m relates the mass of fractal object, M, to its linear size, R, as

$$M = cR^{D_m} \tag{3.2}$$

where c is a constant

D_m is called the fractal dimension (with respect to the mass) estimate for a given object as

$$D_m \approx \frac{\ln M}{\ln R} \tag{3.3}$$

The particle-counting fractal dimension, D_f, relates the number of primary particles N, a fractal object contains with the linear size, R, of that object to the equation

$$N \propto R^{D_f} \tag{3.4}$$

For the calculation of the particle-counting fractal dimension, D_f, square shaped Region of Interest (ROI) with different side length, R, is drawn on the center of the image of fractal growth patterns and the number of the microstructural elements in each ROI is counted. The logarithm of the number of microstructural elements, ln (N(R)), is plotted against the logarithm of the side length of each ROI, ln (R), for varying values of R. The slope of the linear regression curve of this log-log plot is the particle-counting dimension, D_f, (Litwinenko et al., 2002).

3.4.1 Development of a Computer Based Tool for Fractal Dimension Calculation

Some aspects were taken into consideration in the process of development of a computer system in relation to compute the necessary calculations in the present work. It is pre-determined that importance is placed on the calculations of fractal dimension. The study contemplated the practicality and dynamism of the chosen hardware and software platform for the system. Accuracy of the calculations was considered of paramount importance.

The computer application has been developed on a modest system, as consideration was given to the transferability of the application. All the tests to acquire data for works on calculation of the fractal dimension calculation of the experimentally cultured fractals were carried out on the exact same system the application was developed on, to eliminate the probability of achieving inconsistent fractal dimension values (Jelinek and Fernandez, 1998). The system consists of (components stated with relevance to the development of the application) an Intel core i3 processor (2.67 GHz), an Intel Chipset PCH HM57 main

board, 2GB of SD RAM and Graphics Controller. For the software, Visual Basic 6.0 was used.

3.4.2 Image and Image Data Processing

An important factor that has been consistently considered in the present work was the handling of image data (Jain, 1989). In the processing of a digital image, the image needs to be converted into an array of values to maintain the information contained within it. This enables easy manipulation of image information such as in the measurement of structures and the application of image processing techniques for further enhancements to the system.

The processing of the image includes processes such as thresholding, edge detection and contouring. Thresholding was done to highlight points of an image at or near a particular value. For this particular system, each image involved was assumed to have a single range of Hue, Saturation Value (HSV) in their structures and thresholding was done based on the HSV color definition.

Currently, a computer application was developed with the purpose of measuring fractal dimension between 1 and 2. Before the estimation of fractal dimension value can be done, a certain degree of tweaking and/or processing of the images is necessary for the purpose of achieving a kind of suitable ‘environment’ or characteristics in order to achieve relevant fractal dimension values (Ribeiro et al., 2002, Jelinek and Fernandez, 1998, Foroutan-pour et al., 1999).

In the processing of images, the system carried out the elimination of (non-fractal and/or irrelevant) objects. This was done using the 'thresholding' method based on the Hue, Saturation, (luminosity) Value or HSV color definition. Besides the estimation of fractal dimension values of the Koch curves and Sierpinski carpet and gasket, the system was also used to analyze non-linear fractals. Therefore, the said image processing capabilities were deemed necessary.

Edge detection and contouring enables the highlighting of irregular structures within the image(s) (Jain, 1989). The system applied here utilized techniques based on Neighborhood Averaging, accomplished by replacing each pixel with the average of itself and its neighbors. This was generalized by the equation:

$$P'_{x,y} = \frac{\sum_{i,j=-m}^{m} W_{i,j} \cdot P_{x+1,y+1}}{\sum_{i,j=-m}^{-m} W_{i,j}} \tag{3.5}$$

Eq. (3.5) shows the sum of pixel value in a region multiplied by a set of integer weights. The calculation for this equation is performed over a square of dimension $2m + 1$ thus sizes in range of 3×3, 5×5, etc. As the simplest, non-resource-exhaustive method is of importance, only the calculations involving 3×3 neighbourhood are reported in this work. The 3×3 matrices for the edge detection and contouring is:

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

In the contour method, the resultant pixels were biased so that the brightness of pixels is inverted.

3.4.3 Determination of Fractal Dimension

The Box-Count Method (BCM) was implemented in the process of calculation of fractal dimensional values. The method was chosen in this study based on its ease of use, automatically computable and applicable for patterns with or without self-similarity (Foroutan-pour et al., 1999). In this method, a regular mesh grid is overlapped over a (fractal) structure and the amount of boxes containing the structure of an object being analyzed is counted (Milosevic and Ristanovic, 2007). Let the value of the mesh size be s and the corresponding boxes which contain the structure $N(s)$. The box size, s is decreased successively and the corresponding amount of boxes, $N(s)$ is counted. As the size of the box, s decreases, the boxes approximates to the structure.

For a regular structure;

$$N(s) = \text{constant} \times (1/s)^{-D} \tag{3.6}$$

and

$$\text{Log } N(s) = -D \text{ Log } (1/s) + c \tag{3.7}$$

where D is the dimension of the structure. Therefore the values of log $N(s)$ plotted versus log s would produce a linear graph where the gradient of the line represents the fractal dimension value. The box counting method proposes a systematic measurement, which applies to any structure in a 2-dimensional plane and can readily be adapted for structures in a 3-dimensional plane.

The first procedure in the BCM is the application of mesh grid over an image. The grid is generated using parallel lines spaced out evenly in both vertical and horizontal directions. Existing line methods in programming techniques were applied in creating these lines. This resulted in the segmentation of the image(s) into boxes of equal sizes. In the present work, only boxes of the mesh grid containing the structure are considered. The Red, Green, Blue (RGB) value of the pixels of an image is extracted and boxes containing pixels in which the structure is defined are selected. This requires the segmentation of the image by grid boxes. The algorithm for this step is as follows:

```
For <pixels of the i-th grid box>
    For pixels 1 to box height
        For pixels 1 to box width
        If the current pixel (x,y) represents the structure
        then add box count and go to next box i+1
```

Images will continuously be scanned over several box sizes (set by the user), resulting in the box count for the grid boxes containing the structure within the image. However, the suitability of the color definition of RGB in the analysis of the images was also taken into consideration. Since the development of the system also considered its ability to later analyze natural images (that may vary within a certain range of shades), a better representation of the colour of natural structures was taken into account. A suitable method for this representation is the Hue, Saturation Value (HSV) model (Bhattacharya et al., 1996). For the BCM that was implemented in the system, only the structures defined within the range given by the selected HSV were considered.

As shown in Figure 3.6, the main window of the fractal dimension calculation program developed in this research work consists of three panels: the control panel on the left and upper right, the data panel on the bottom right and the display panel in the middle.

The control panel contains all the buttons and drop down menu to assign the values of the parameters for the fractal image processing and fractal dimension analysis. The data panel presents the box count for the corresponding box sizes in the calculation of fractal dimension. While the 2D fractal image of the experimentally cultured fractals can be showcased on the display panel.

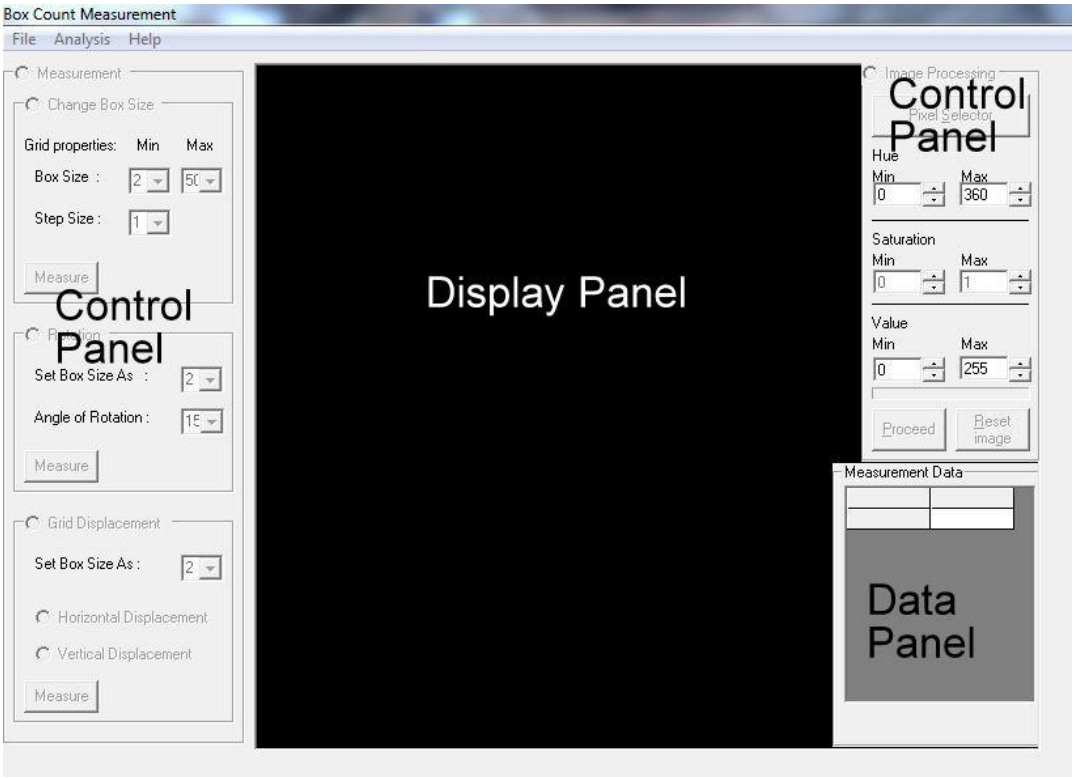


Figure 3.6: The main window of the fractal dimension calculation program

The fractal images were acquired via a digital camera and were later pre-processed via an (of-the-shelf) image processing software before being uploaded to the system. The digital images were scanned into binary images in which an image was segmented to the smallest element, a pixel which is defined by the RGB color definition. In the processing of a digital image, the image was converted into an array of values to maintain the information contained within it. This enabled easy manipulation of image information such as in the measurement of structures and the application of image processing techniques which involved thresholding as well as edge detection and contouring.

A mesh grid that can be generated by using parallel lines spaced out evenly in both vertical and horizontal directions was overlapped over the image. The purpose of applying a mesh grid was to segment an image into boxes of equal size as those given by the mesh grid. Only the boxes of the mesh grid containing the structure were considered. As the analysis was done upon digitized images, the information contained by pixels was utilized to allow the structure to be measured. The RGB value of the pixels of an image was extracted and the boxes containing pixels which defined the structure were selected. Moreover, only the boxes containing the structure defined within the range given by the selected Hue Saturation Value (HSV) were considered.

In the data acquisition, different box sizes were used for measuring fractal figures. The estimated measurement values of the fractal figures were dependent on measurement sampling. The confidence in the estimate of the fractal dimension value can be strengthened by increasing the number of measurements. The calculation of fractal

dimension value was achieved by statistical inference of the linear regression model (Zhou et al., 2008).

The fractal dimension value determined is based on the gradient of the least squares regression line. Based on the values to be considered for the calculation of the gradient (fractal dimension value), a linear regression line was drawn upon the plot of the log-log graph of the number of boxes counted, $N(s)$ versus the corresponding box size, s.

C H A P T E R 4

EXPERIMENTALLY CULTURED FRACTALS

This chapter presents the results and various features of the cultured fractals. Characterization of the original fractals observed in polymer films was done and their digital images were taken and analyzed.

4.1 Growth Patterns

Fractal aggregates especially diffusion-limited aggregate involve the random walk of particles and their subsequent sticking (Chandra and Chandra, 1994). To obtain fractal aggregates in laboratory framework, a system with particles in random walk is required. In polymer electrolytes, polymers added with inorganic salts, the anions as well as the cations are found to be mobile (Chandra, 1996, Hashim Ali et al., 2000). Therefore, they can be considered as a natural framework for fractal growth.

In the present work, fractals were obtained in laboratory conditions using solid polymer electrolyte films as the media of growth. Films of chitosan-silver nitrate (chitosan-$AgNO_3$), polyethylene oxide-ammonium iodide (PEO-NH_4I) and poly (vinylidene fluoride-co-hexafluoropropylene)/poly (ethyl methacrylate)-ammonium trifluoromethanesulfonate (PVDF-HFP/ PEMA-$NH_4CF_3SO_3$) containing various wt% of chromium oxide (Cr_2O_3) were prepared via solution casting method. In these particular systems, the movement of ions (anions and cations) from the inorganic salts was expected. The formation of fractals was believed to be due to the ions from the inorganic salt performing random walks and their subsequent sticking (Chattaraj et al., 1996,

Chandra, 1996, Somasundaran and Runkana, 2003). The important experimental feature is that no external bias was necessary for either the creation or subsequent aggregation of the 'random walkers/ions' as done in the electrodeposition systems (Chandra, 1996).

4.1.1 Chitosan-$AgNO_3$ Films as Media to Culture Fractals

As mentioned earlier, chitosan-$AgNO_3$ films were employed as one type of the media to culture fractals. Out of all the successfully cultured fractals, only samples containing visibly clear fractal patterns were chosen for analysis. The fractal aggregates of different sizes in the chitosan-silver nitrate film are shown in Figure 4.1. Images of the laboratory-formed fractals were later acquired in order to proceed with the fractal dimension estimation. The images were acquired via a digital camera.

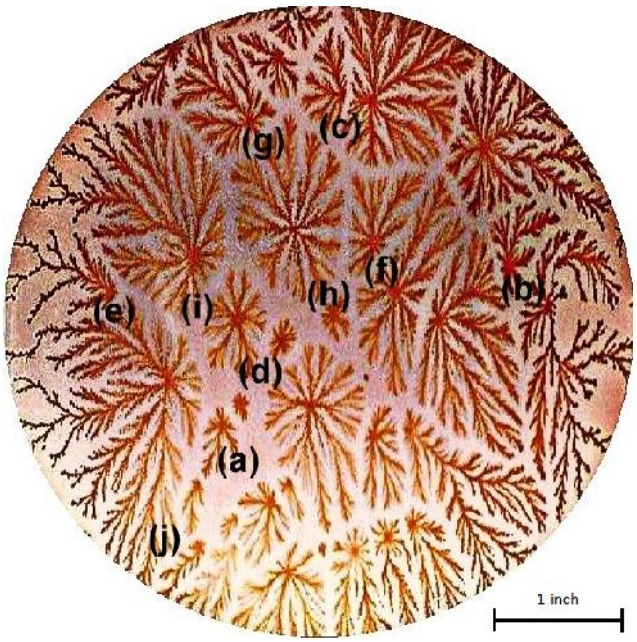


Figure 4.1: Fractals in film of chitosan added with silver nitrate

As can be observed in Figure 4.1, fractals formed at different nucleation centers and then grew in the direction away from the nucleation sites. The fractals grew irregularly and in an unpredictable motion which can be attributed to the Brownian motion of aggregating species (Arof and Mohamed, 1995). Consequently, the size, radius of cluster and number of branches were implemented as fractal parameters. The different sizes and shapes of these fractals are due to the occurrence of other nucleation sites. An interesting feature of the fractals is that they do not overlap each other but are separated by a definite boundary.

4.1.2 PEO-NH_4I Films as Media to Culture Fractals

Another type of medium used for culturing of fractals was PEO-NH_4I system. A number of petri dishes of cultured samples have been prepared in the laboratory with different amounts of PEO:NH_4I weight ratios of 95:5, 90:10, 85:15, 80:20, 75:25, 70:30, 65:35, 60:40, 55:45 and 50:50 with 3 samples from every weight ratio. Fractal patterns were fairly obvious in the samples with weight ratios of PEO: NH_4I of 70:30 and 60:40. From close observation, samples with weight ration 60:40 were selected for better fractal analysis. Figure 4.2 depicts fractals observed in one of the samples of the PEO: NH_4I of 60:40 weight ratios.

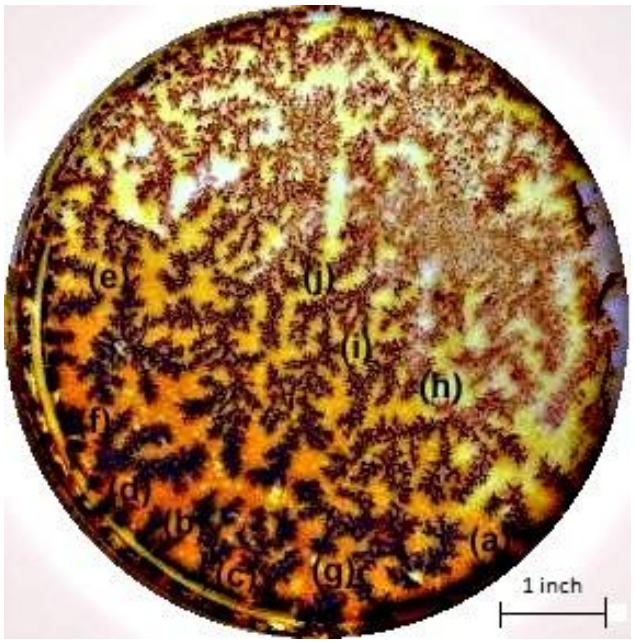


Figure 4.2: Digital images of the fractal patterns observed in PEO-NH_4I films

The PEO-NH_4I films are ion conducting polymer films permitting both cations and anions to move. According to Chandra and Chandra (1994), the fractal aggregates observed in such films are principally due to the random walk and aggregation of iodine. Similar to the fractals observed in chitosan-$AgNO_3$ films, the fractals are fern-like aggregates consisting of different number of branches extending from their respective nucleation center.

The different sizes and shape of these fractals are due to the occurrence of other nucleation sites. The largest cluster was found to be 5 cm (2 inches) in length. For simulation purposes, the images of the fractal patterns displayed in Figure 4.2 were chosen specifically. Obviously, as the fractals become bigger, their density, branching and sizes also increase.

4.1.3 PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3 Films as Media to Culture Fractals

PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films were also used as media for culturing fractals. There were a couple of films that managed to produce fractals with apparent appearance. The addition of Cr_2O_3 caused the formation of greenish fractals. Shown in Figure 4.3, are the fractal aggregates of different sizes that have been obtained in the films of PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 without any external stimuli. Evidently, fractals were found to form in the film in various sizes, density and branching.

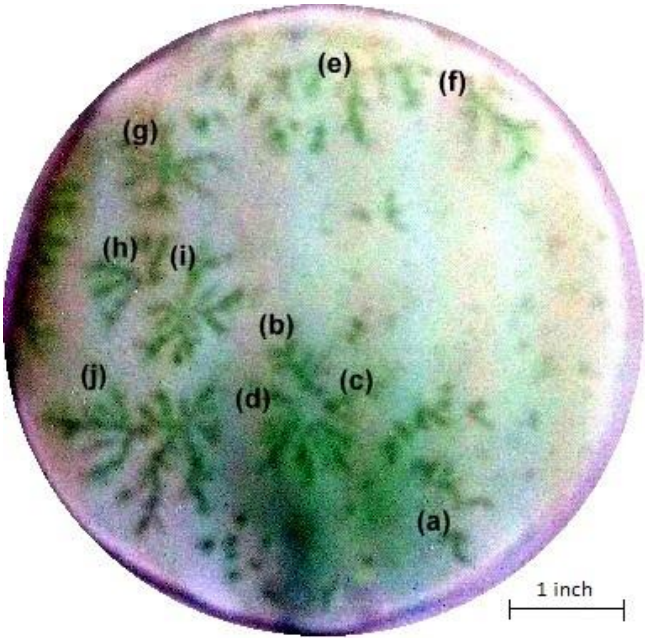


Figure 4.3: Fractals in films of PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3

The PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films can be classified as polymer electrolytes. The PVDF-HFP/PEMA act as hosts while the ammonium salts dissociate in them to provide the ions necessary for conduction. During the first couple of weeks of drying time, it was detected that aggregates started to form in the films of ~0.05mm in thickness. Consequently, these aggregates showed ramified growths resulting in branched

morphologies. Within a few months, the ramified growths stopped and full grown fractals were discovered. The fractals were observed to grow from nucleation centers in a certain direction away from the nucleation site and separated from each other by a definite boundary. An explanation on why the polymer behaves like this could be that in most polymer electrolytes the anions as well as the cations are found to be mobile. It is in this context that polymer electrolytes can be considered as natural framework for the growth of fractals due to random walk of free anions or cations (Chandra and Chandra, 1994).

The fractals are principally due to random walkers and their subsequent aggregation. The different sizes and shapes of these fractals are due to the occurrence of other nucleation sites. The largest cluster was found to be 2.5 cm in length. The resulting clusters were mostly branched. This was possibly due to the fact that the arriving particles were far more likely to stick to the tips of outer branches than to maneuver their way deep into the fjords before contacting the surrounding branches. Thus, the tall branches of the cluster screen the small ones and grew faster. The growth on the tips, however, was not always in the outward radial direction. Sometimes a few new branches were spun off from one tip site as occurred in the original seed. The tip-splitting makes the DLA clusters a self-similar fractal.

4.2 Fractal Dimension of Cultured Fractals

After the fractals have been obtained, it is important to measure their dimension. The fractal dimension actually measures the compactness of a fractal aggregate that is, how it fills space. In order to calculate the fractal dimension of a chosen image, a digital image of the fractal must be acquired. This image was then processed and uploaded into

the fractal dimension measuring program created specifically for this purpose. The grid lengths of square meshes are taken as integer values from 1 to 50 by using fixed step size of 1 unit. For every film shown in Fig. 4.1 to 4.3, ten fractal clusters labeled (a) to (j) were chosen for this calculation purpose.

In the fractal dimension measuring program, after the digital image of the fractal was uploaded, the image processing button was selected (Suki et al., 2007). Here, one can adjust the level of hue and saturation of the image for scanning. After scanning the image, one can check the measurement button and set the parameters for box size and grid displacement for image processing. At this stage, the program will start to calculate the numbers of box count for different box sizes and thus the result of the fractal dimension values can be obtained by clicking the analysis button on the upper left menu of the program (Figure 4.4).

As shown in Figure 4.4(a)(i)-(iii), the images of the fractal patterns of Figure 4.1(j), Figure 4.2(j) and Figure 4.3(j) respectively were scanned for image processing. These images were converted into binary images before the overlapping of mesh grid was done, Figure 4.5(b)(i) -(iii).

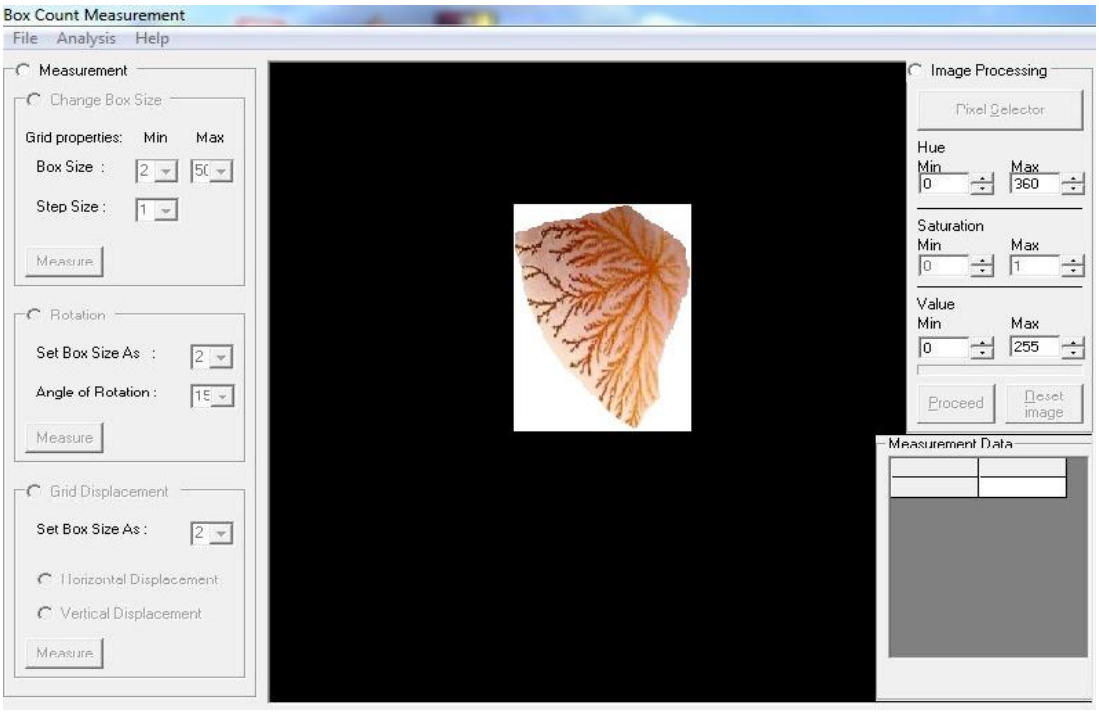


Figure 4.4(a) i: Screen shot image of the box-count for the experimentally obtained fractal aggregate shown in Figure 4.1(j) being uploaded for image processing

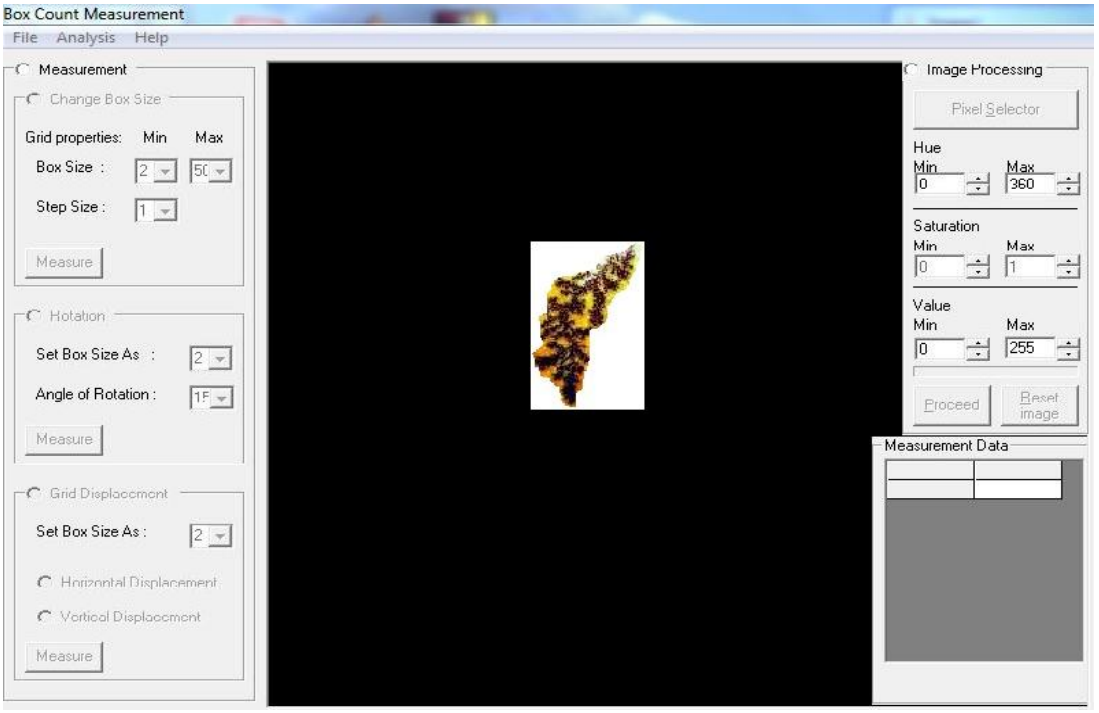


Figure 4.4(a) ii: Screen shot image of the box-count for the experimentally obtained fractal aggregate shown in Figure 4.2(j) being uploaded for image processing

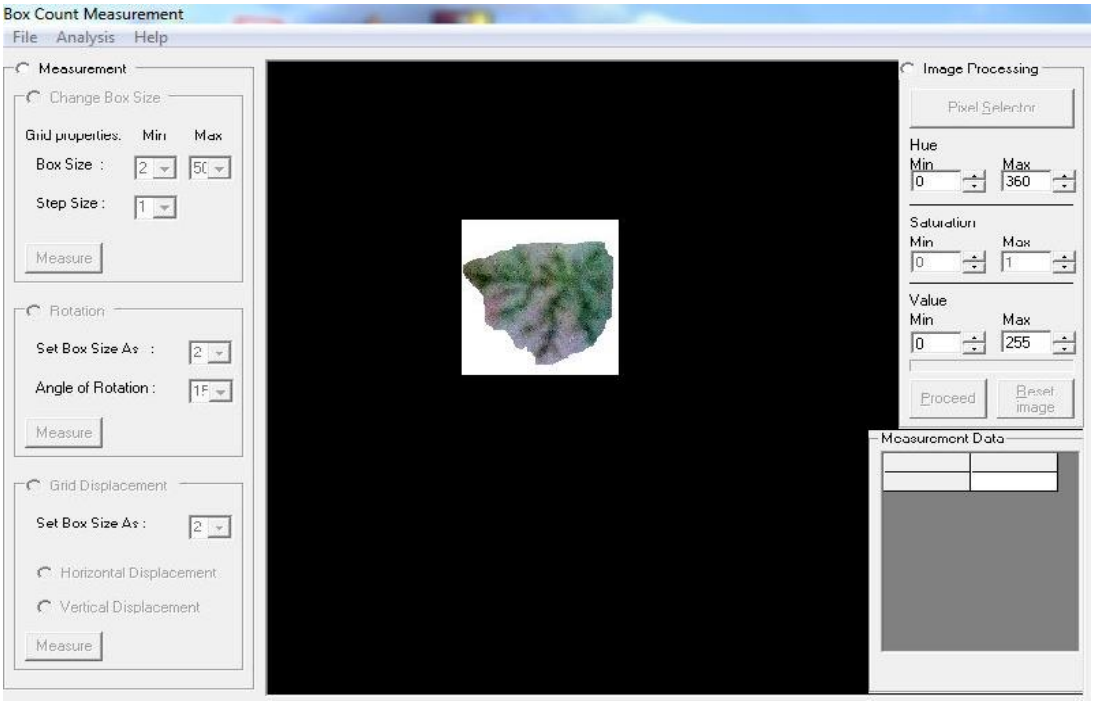


Figure 4.4(a) iii: Screen shot image of the box-count for the experimentally obtained fractal aggregate shown in Figure 4.3(j) being uploaded for image processing

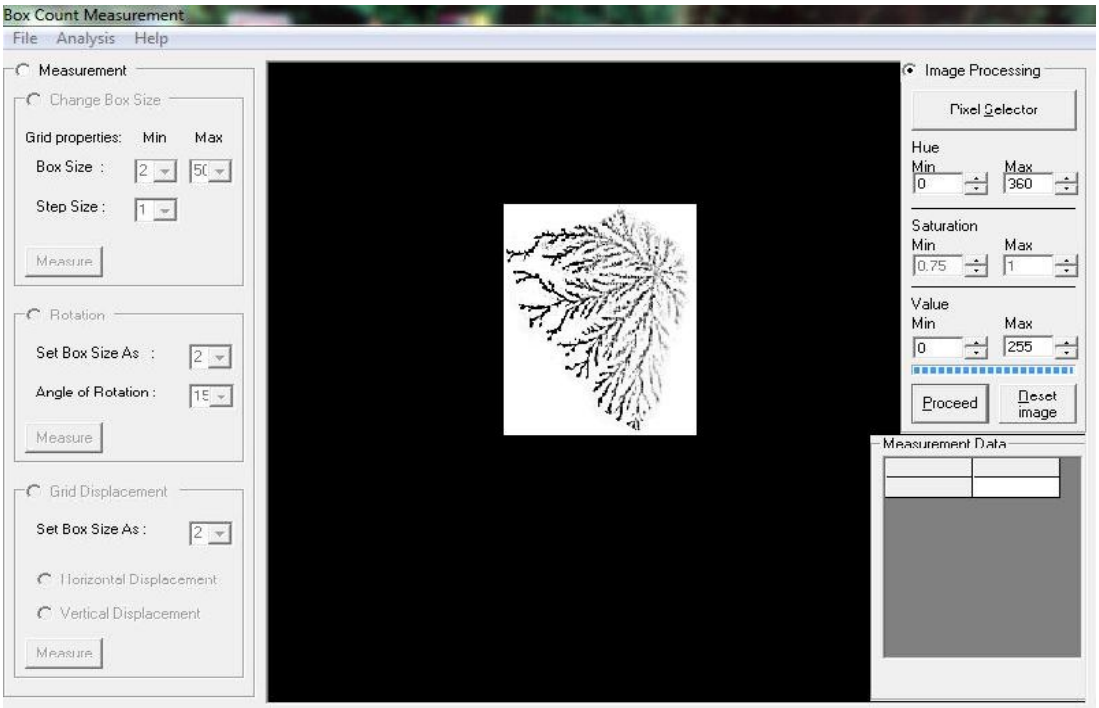


Figure 4.4(b) i: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.1(j) being converted into binary image

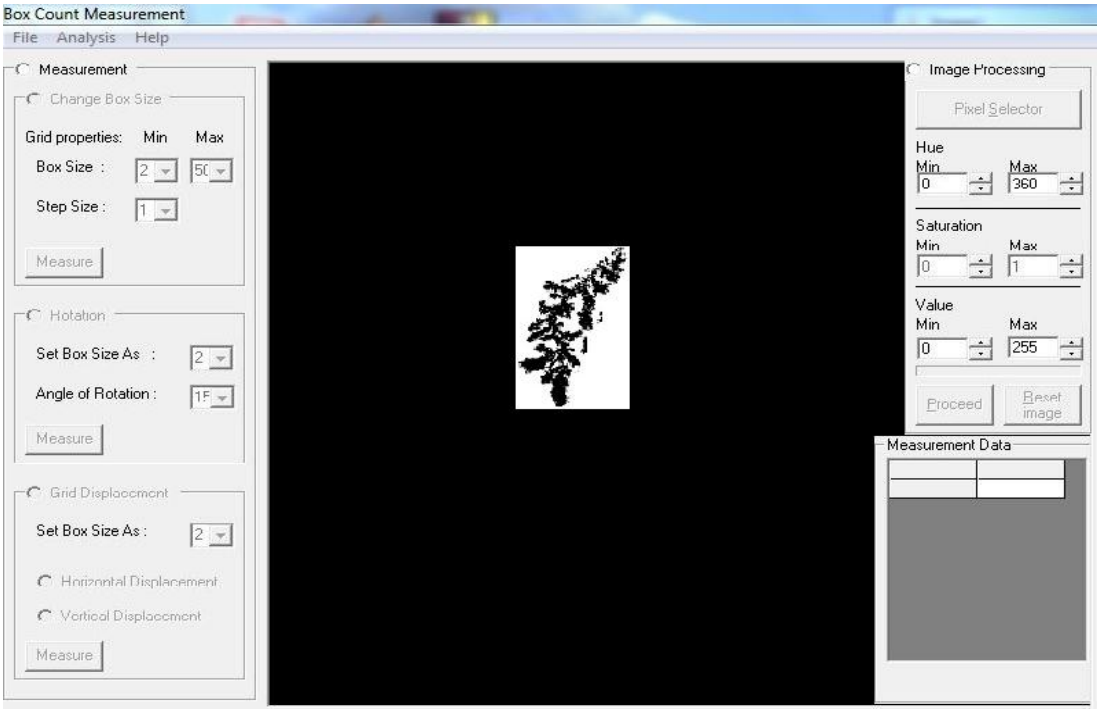


Figure 4.4(b) ii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.2(j) being converted into binary image

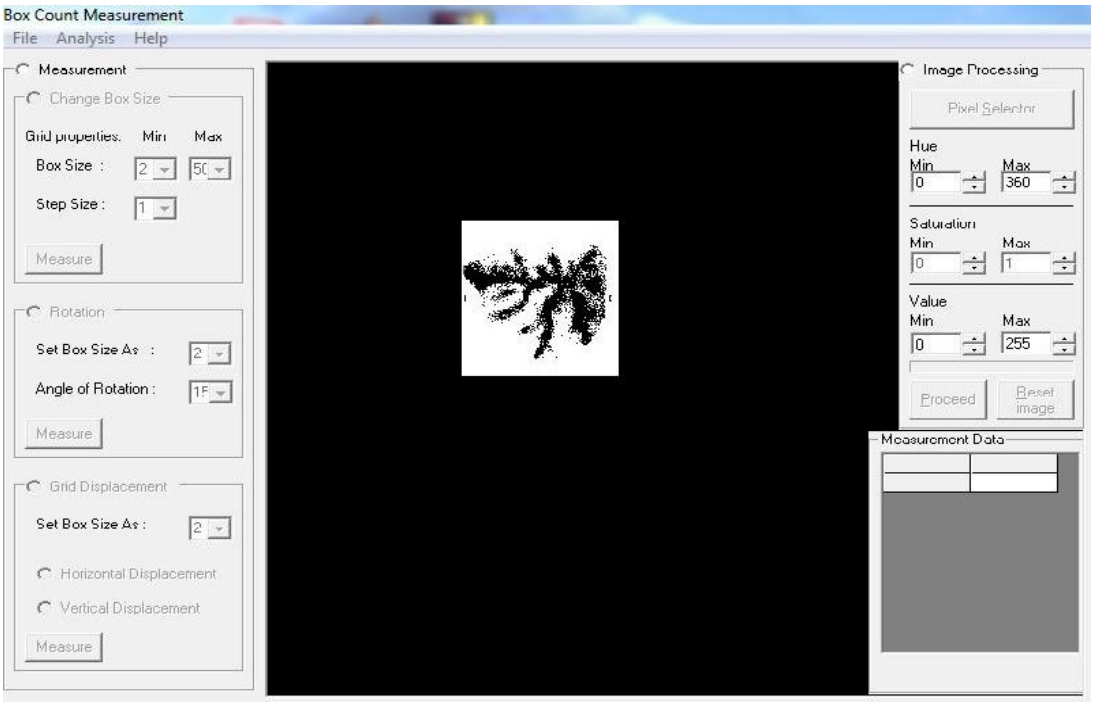


Figure 4.4(b) iii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.3(j) being converted into binary image

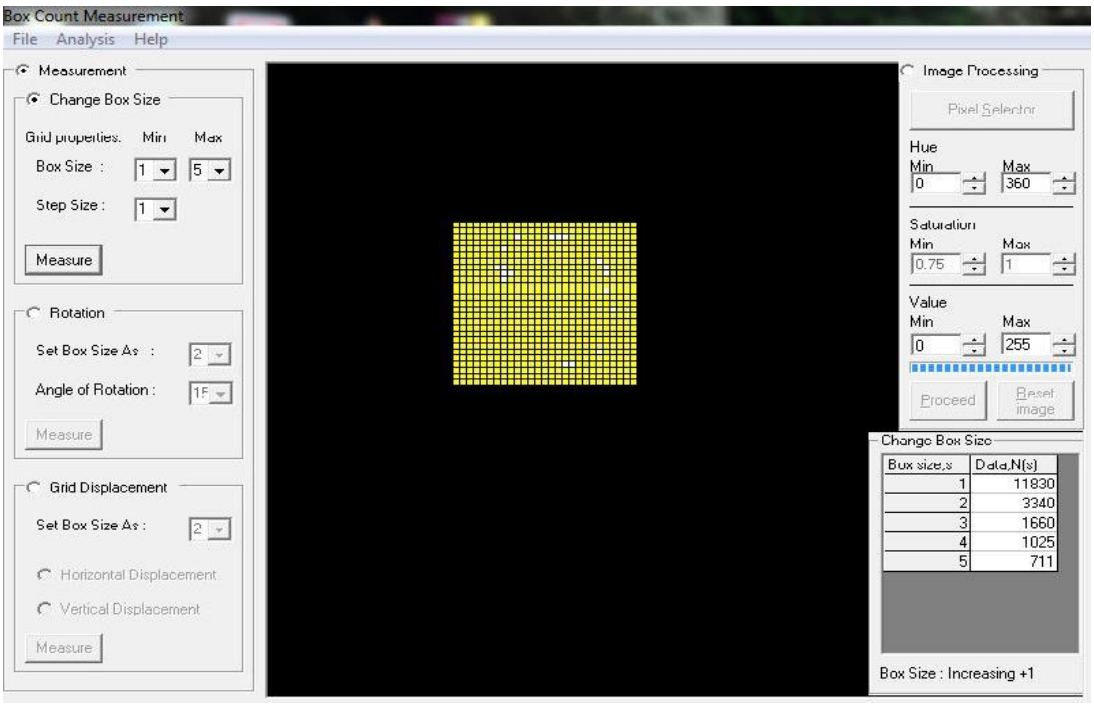


Figure 4.4(c) i: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.1(j) being overlapped by mesh grid of box size 5

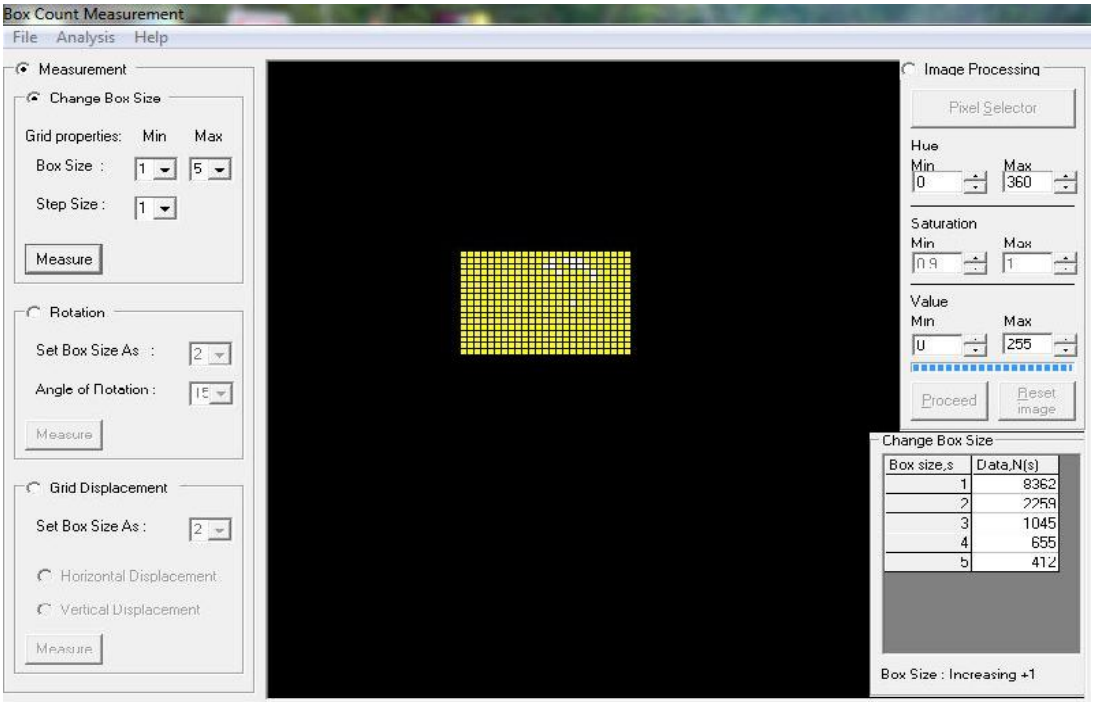


Figure 4.4(c) ii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.2(j) being overlapped by mesh grid of box size 5

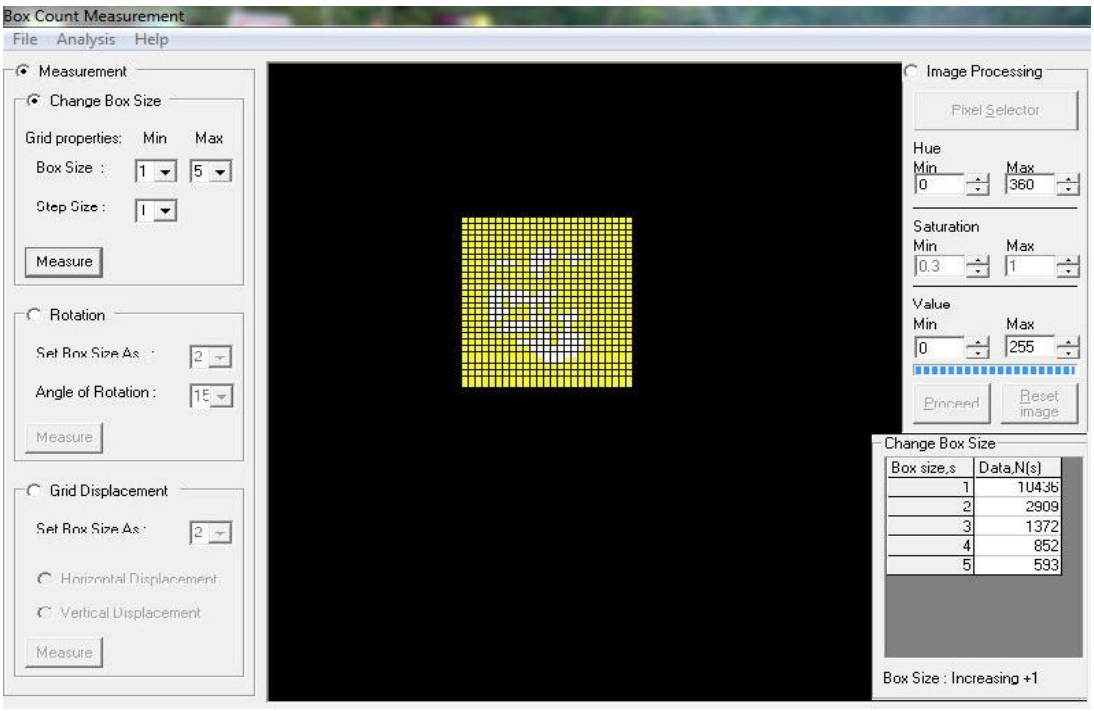


Figure 4.4(c) iii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.3(j) being overlapped by mesh grid of box size 5

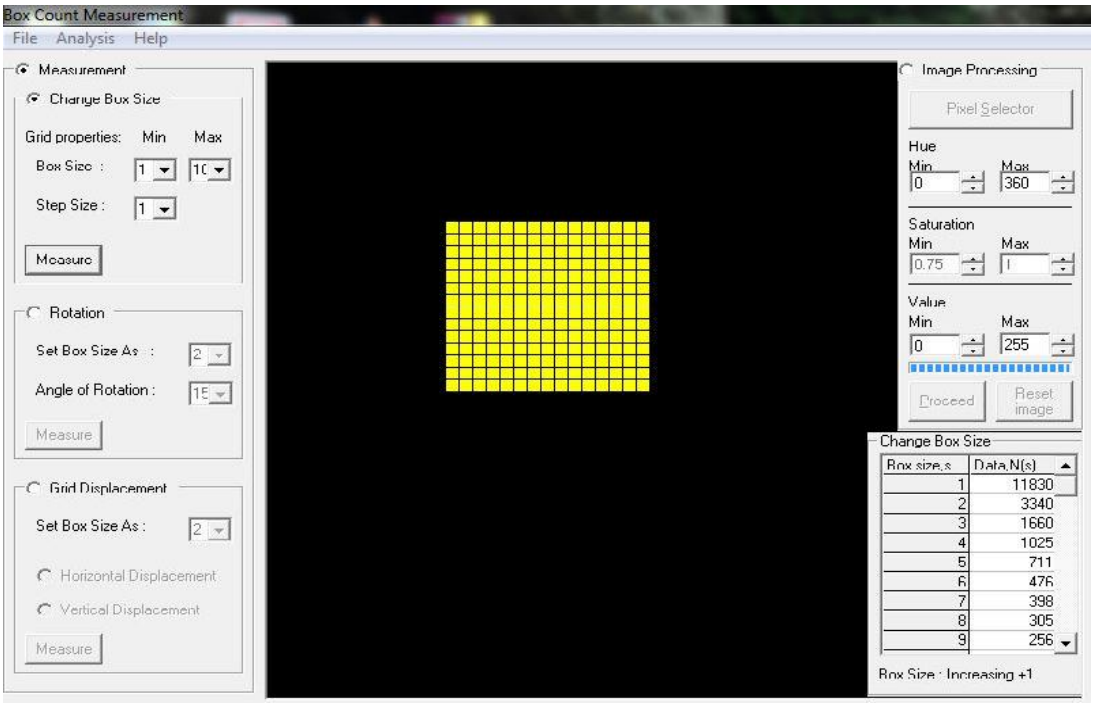


Figure 4.4(d) i: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.1(j) being overlapped by mesh grid of box size 10

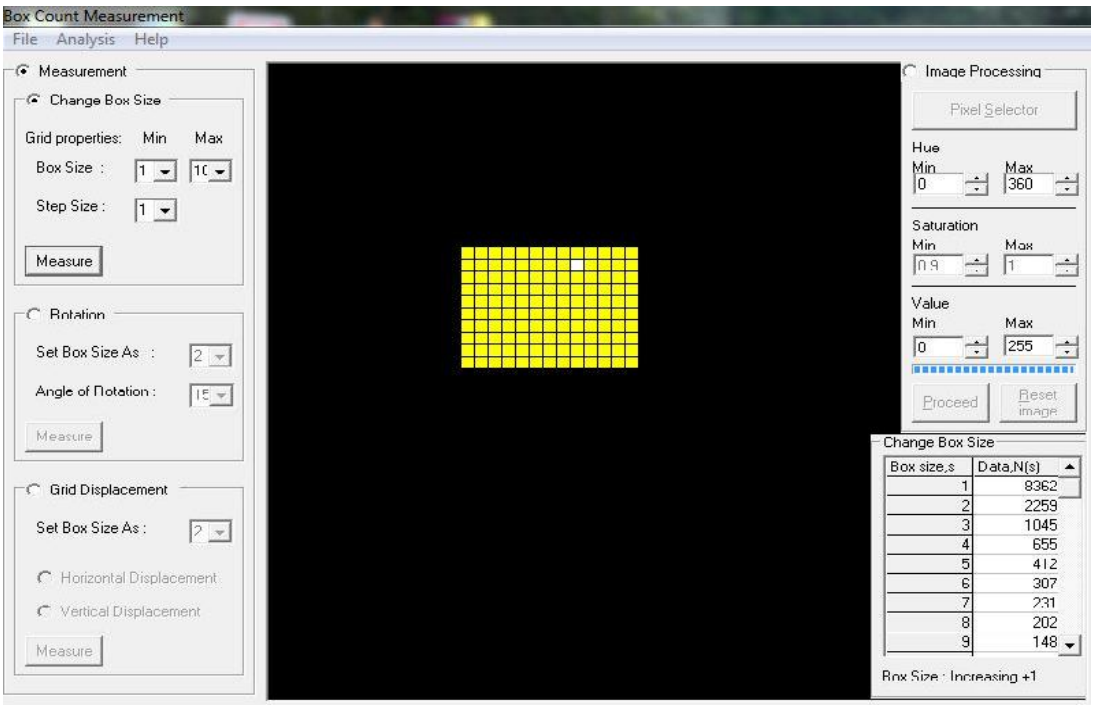


Figure 4.4(d) ii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.2(j) being overlapped by mesh grid of box size 10

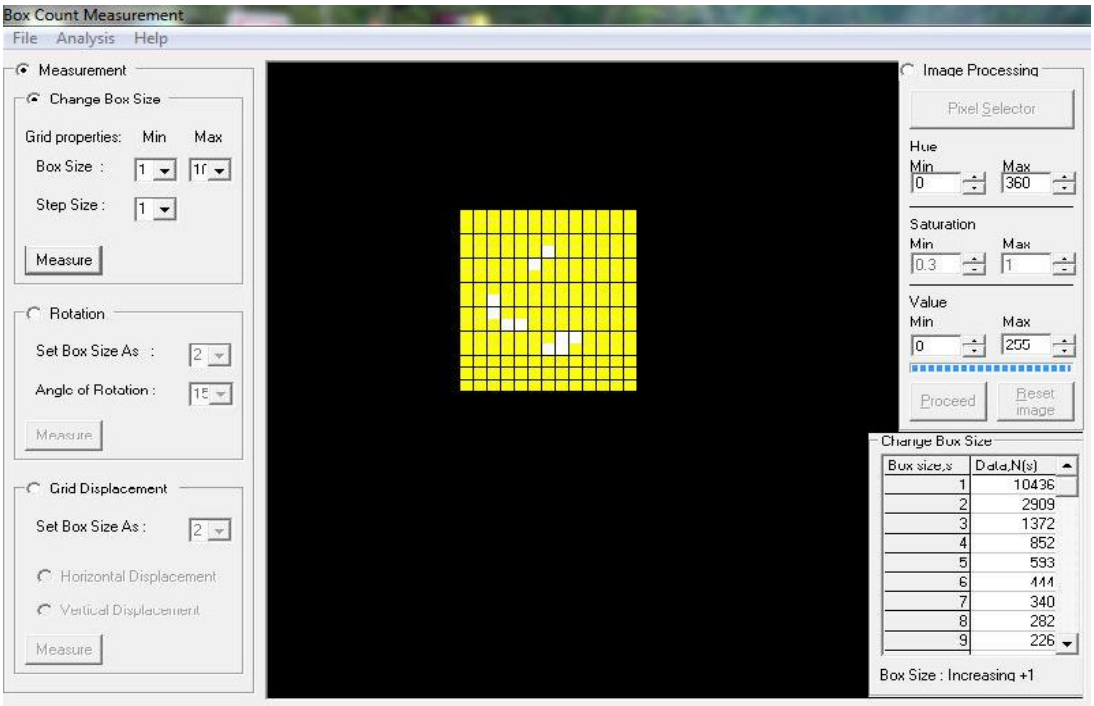


Figure 4.4(d) iii: Screen shot image of the image of the experimentally obtained fractal aggregate shown in Figure 4.3(j) being overlapped by mesh grid of box size 10

Figure 4.4(c) (i) - (iii) show that the box-count measurement was done by overlapping a box with size 5 and in Figure 4.4(d) (i) - (iii) are for box size 10. The measurement procedure started with box size 1 and was repeated until the final box size of 50. The collected box-count data for the image of Figure 4.1(j) are listed in Table 4.1.

Table 4. 1: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (j)

Box size,s	N(s)	Log (s)	Log N(s)	Box size,s	N(s)	Log (s)	Log N(s)
1	10939	0	4.0390	26	48	1.415	1.6801
2	3017	0.301	3.4796	27	45	1.4314	1.6528
3	1417	0.4771	3.1514	28	42	1.4472	1.6264
4	870	0.6021	2.9395	29	40	1.4624	1.6010
5	743	0.699	2.8710	30	38	1.4771	1.5765
6	548	0.7782	2.7387	31	36	1.4814	1.5528
7	423	0.8451	2.6268	32	34	1.5051	1.5298
8	339	0.8031	2.5299	33	32	1.5185	1.5075
9	278	0.9542	2.4445	34	31	1.5315	1.4859
10	233	1	2.3680	35	29	1.5441	1.4649
11	199	1.0414	2.2988	36	28	1.5563	1.4445
12	172	1.0782	2.2357	37	27	1.5682	1.4247
13	151	1.1138	2.1776	38	25	1.5798	1.4053
14	133	1.1481	2.1238	39	24	1.5811	1.3865
15	119	1.1781	2.0738	40	23	1.6021	1.3682
16	106	1.2041	2.0269	41	22	1.6128	1.3503
17	96	1.2304	1.9829	42	22	1.6232	1.3329
18	87	1.2553	1.9414	43	21	1.6335	1.3158
19	80	1.2788	1.9022	44	20	1.6435	1.2992
20	73	1.301	1.8650	45	19	1.6532	1.2829
21	68	1.3222	1.8296	46	18	1.6628	1.2670
22	62	1.3424	1.7958	47	18	1.6721	1.2514
23	58	1.3617	1.7636	48	17	1.6812	1.2362
24	54	1.3802	1.7327	49	17	1.6902	1.2213
25	50	1.3878	1.7030	50	16	1.699	1.2066

The fractal (box) dimension was determined by taking the value of the slope of the linear fit line of the plotted graph of Log $N(s)$ versus Log s as shown in Figure 4.5(a).

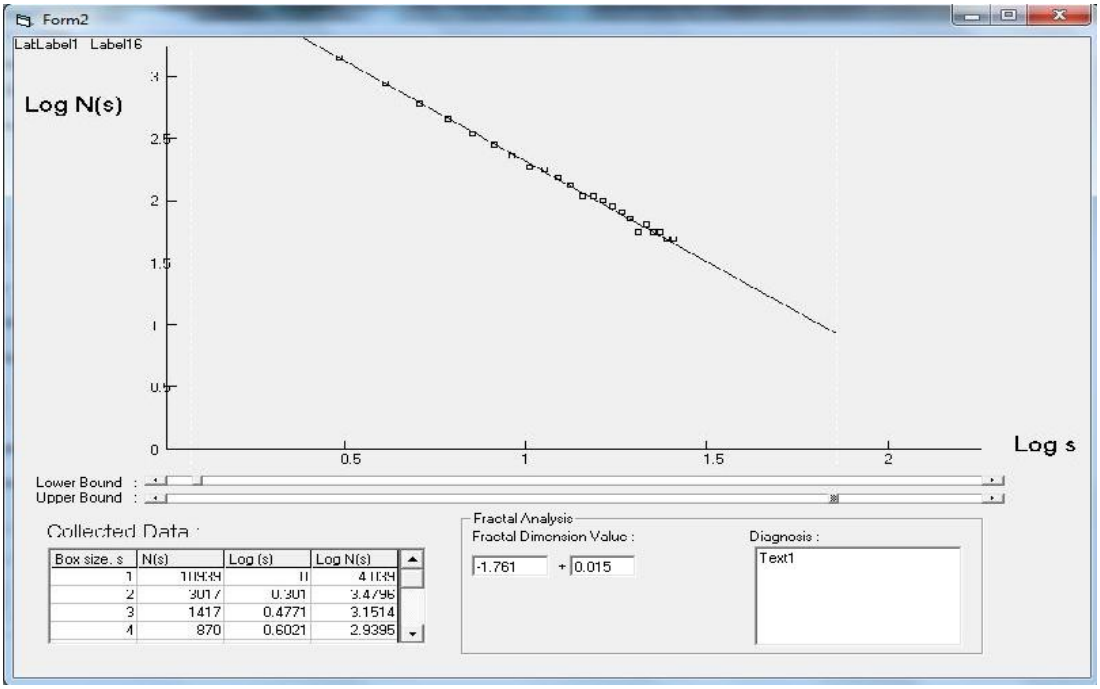


Figure 4.5(a): Screen shot of the graph of log $N(s)$ vs. log s (linear scale) withthe calculated fractal (box) dimension, D = 1.761 for the image of Figure 4.1(j)

Table 4.2 gives the collected box-count data followed by the graph of the fractal dimension calculation for the image of Figure 4.2(j) as depicted in Figure 4.5(b).

Table 4.2: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (j)

Box size,s	N(s)	Log (s)	Log N(s)
1	9322	0	3.9695
2	2446	0.301	3.3885
3	1118	0.4771	3.0484
4	681	0.6021	2.8331
5	519	0.699	2.7155
8	374	0.7782	2.5734
7	284	0.8451	2.4533
8	223	0.8031	2.3493
9	181	0.9542	2.2575
10	150	1	2.1754
11	126	1.0414	2.1011
12	108	1.0782	2.0333
13	94	1.1138	1.9710
14	82	1.1481	1.9132
15	72	1.1781	1.8595
16	64	1.2041	1.8092
17	58	1.2304	1.7620
18	52	1.2553	1.7174
19	47	1.2788	1.6753
20	43	1.301	1.6353
21	40	1.3222	1.5973
22	36	1.3424	1.5611
23	34	1.3617	1.5264
24	31	1.3802	1.4933
25	29	1.3878	1.4615

Box size,s	N(s)	Log (s)	Log N(s)
26	27	1.415	1.4310
27	25	1.4314	1.4016
28	24	1.4472	1.3802
29	22	1.4624	1.3424
30	21	1.4771	1.3222
31	20	1.4914	1.3010
32	19	1.5051	1.2693
33	18	1.5185	1.2453
34	17	1.5315	1.2220
35	16	1.5441	1.1994
36	15	1.5563	1.1775
37	14	1.5682	1.1561
38	14	1.5798	1.1354
39	13	1.5811	1.1151
40	12	1.6021	1.0954
41	12	1.6128	1.0762
42	11	1.6232	1.0574
43	11	1.6335	1.0391
44	10	1.6435	1.0211
45	10	1.6532	1.0036
46	10	1.6628	0.9865
47	9	1.6721	0.9698
48	9	1.6812	0.9534
49	9	1.6902	0.9373
50	8	1.699	0.9215

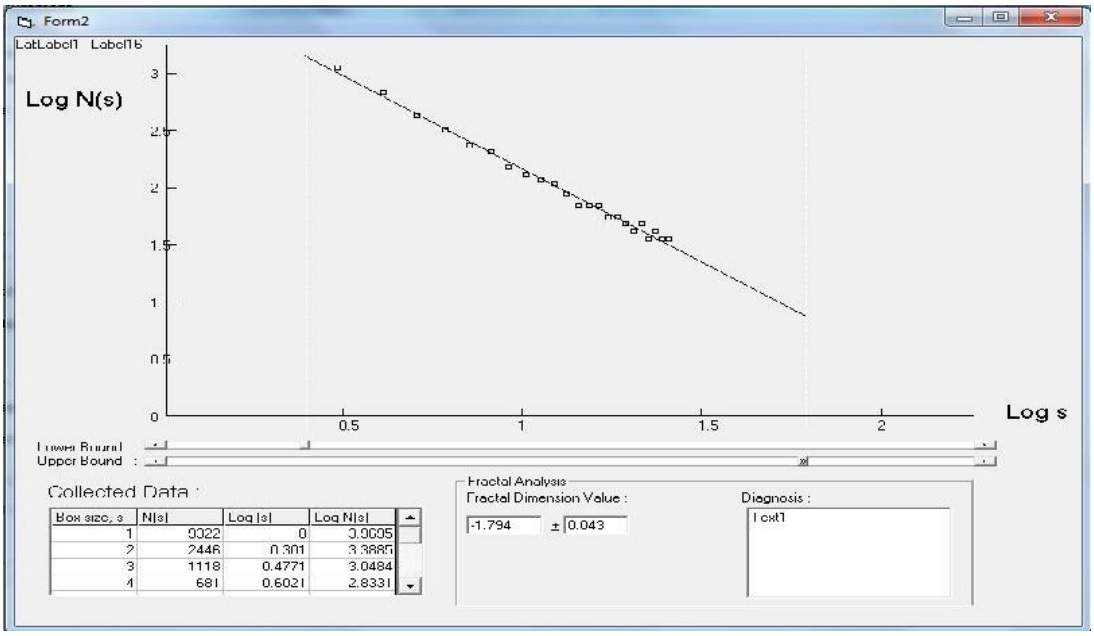


Figure 4.5(b): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.794 for the image of Figure 4.2(j)

The graph of the fractal dimension calculation for the image of Figure 4.3(j) is depicted in Figure 4.5(c) followed by Table 4.3 which gives the collected box-count data.

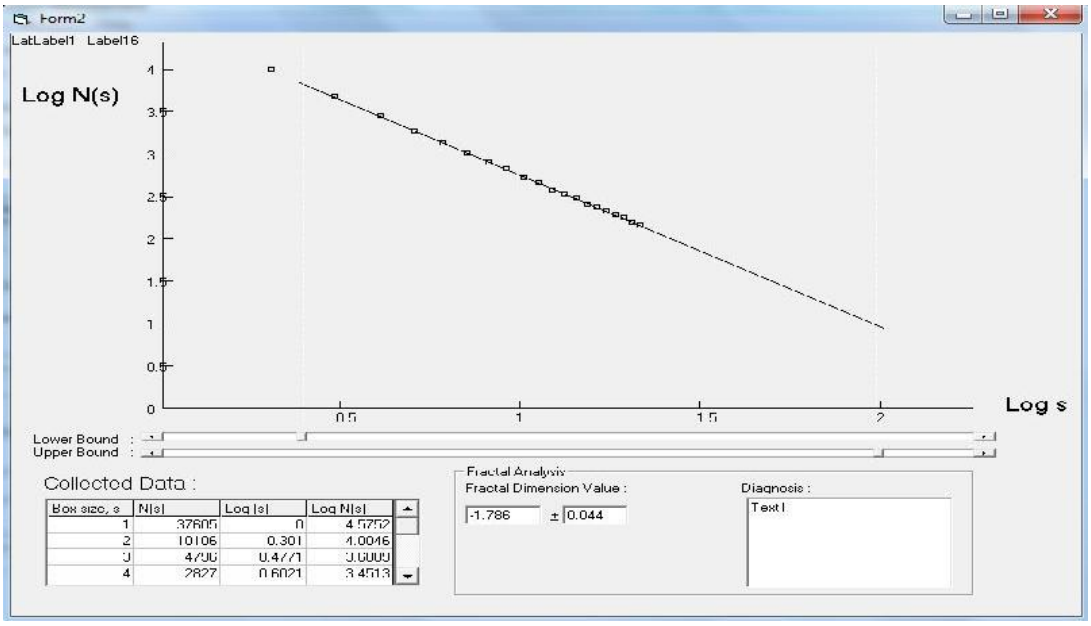


Figure 4.5(c): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.786 for the image of Figure 4.3(j)

Table 4.3: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (j)

Box size,s	N(s)	Log (s)	Log N(s)
1	37605	0	4.5752
2	10106	0.301	4.0046
3	4796	0.4771	3.6809
4	2827	0.6021	3.4513
5	2122	0.699	3.3268
8	1533	0.7782	3.1854
7	1164	0.8451	3.0659
8	917	0.8031	2.9623
8	743	0.9542	2.8709
10	615	1	2.7892
11	519	1.0414	2.7153
12	444	1.0782	2.6478
13	385	1.1138	2.5857
14	337	1.1481	2.5282
15	298	1.1781	2.4747
16	266	1.2041	2.4246
17	239	1.2304	2.3776
18	215	1.2553	2.3333
19	196	1.2788	2.2913
20	178	1.301	2.2516
21	164	1.3222	2.2137
22	151	1.3424	2.1776
23	139	1.3617	2.1432
24	129	1.3802	2.1101
25	120	1.3878	2.0785

Box size,s	N(s)	Log (s)	Log N(s)
26	112	1.415	2.0481
27	104	1.4314	2.0188
28	98	1.4472	1.9906
29	92	1.4624	1.9634
30	87	1.4771	1.9371
31	82	1.4814	1.9116
32	77	1.5051	1.8870
33	73	1.5185	1.8631
34	69	1.5315	1.8400
35	66	1.5441	1.8175
36	62	1.5563	1.7956
37	59	1.5682	1.7744
38	57	1.5798	1.7537
39	54	1.5811	1.7336
40	52	1.6021	1.7139
41	50	1.6128	1.6948
42	47	1.6232	1.6761
43	45	1.6335	1.6578
44	44	1.6435	1.6400
45	42	1.6532	1.6226
46	40	1.6628	1.6055
47	39	1.6721	1.5888
48	37	1.6812	1.5725
49	36	1.6902	1.5565
50	35	1.699	1.5408

The data for box count dimension of the images of Figure 4.1(a) to (i) are tabulated in Table 4.4 to 4.12 with their screen shots of the graphs which yield their fractal dimension values in Figure 4.6(a) to (i) respectively.

Table 4.4: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (a)

Box size,s	N(s)	Log (s)	Log N(s)
1	5830	0	3.7657
2	1512	0.301	3.1796
3	722	0.4771	2.8585
4	435	0.6021	2.6385
5	375	0.699	2.5740
8	275	0.7782	2.4390
7	211	0.8451	2.3248
8	168	0.8031	2.2259
8	138	0.9542	2.1387
10	115	1	2.0607
11	98	1.0414	1.9901
12	84	1.0782	1.9257
13	74	1.1138	1.8664
14	65	1.1481	1.8116
15	58	1.1781	1.7605
16	52	1.2041	1.7127
17	47	1.2304	1.6678
18	42	1.2553	1.6255
19	38	1.2788	1.5854
20	35	1.301	1.5474
21	32	1.3222	1.5113
22	30	1.3424	1.4769
23	28	1.3617	1.4440
24	26	1.3802	1.4124
25	24	1.3878	1.3822

Box size,s	N(s)	Log (s)	Log N(s)
26	23	1.415	1.3532
27	21	1.4314	1.3252
28	20	1.4472	1.2983
29	19	1.4624	1.2723
30	18	1.4771	1.2472
31	17	1.4814	1.2229
32	16	1.5051	1.1994
33	15	1.5185	1.1766
34	14	1.5315	1.1545
35	14	1.5441	1.1331
36	13	1.5563	1.1122
37	12	1.5682	1.0919
38	12	1.5798	1.0722
39	11	1.5811	1.0529
40	11	1.6021	1.0342
41	10	1.6128	1.0159
42	10	1.6232	0.9981
43	10	1.6335	0.9806
44	9	1.6435	0.9636
45	9	1.6532	0.9470
46	9	1.6628	0.9307
47	8	1.6721	0.9148
48	8	1.6812	0.8992
49	8	1.6902	0.8839
50	7	1.699	0.8690

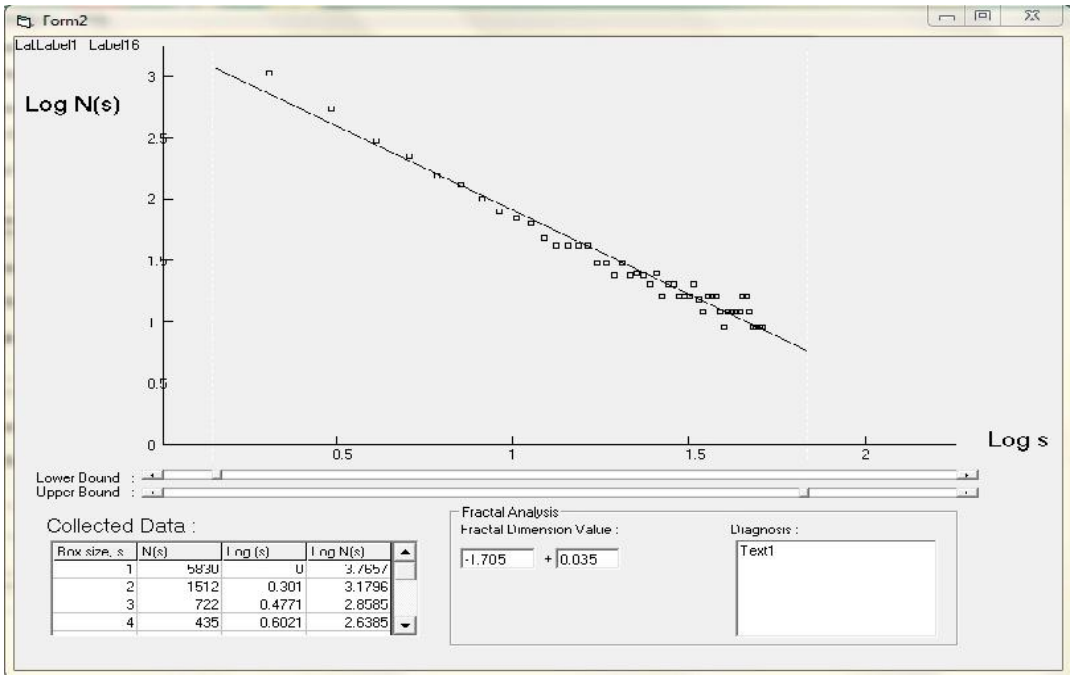


Figure 4.6(a): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.705 for the image of Figure 4.1(a)

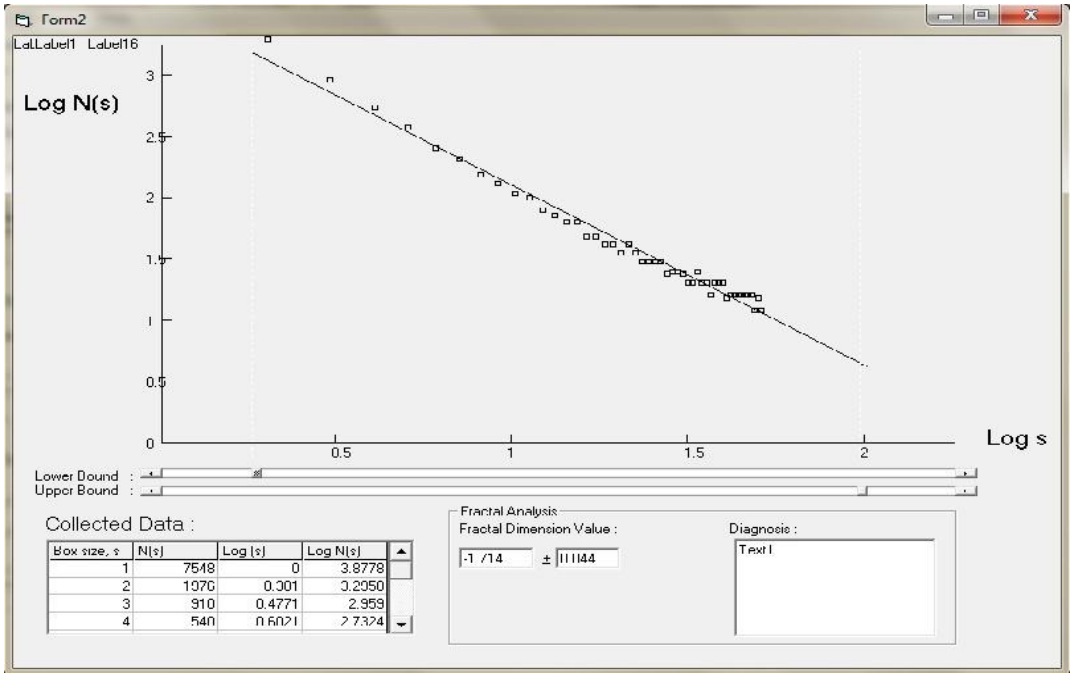


Figure 4.6(b): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.714 for the image of Figure 4.1(b)

Table 4.5: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (b)

Box size,s	N(s)	Log (s)	Log N(s)
1	7548	0	3.8778
2	1976	0.301	3.2958
3	910	0.4771	2.9590
4	540	0.6021	2.7324
5	478	0.699	2.6798
8	350	0.7782	2.5440
7	269	0.8451	2.4293
8	214	0.8031	2.3299
8	175	0.9542	2.2422
10	146	1	2.1638
11	124	1.0414	2.0929
12	107	1.0782	2.0281
13	93	1.1138	1.9685
14	82	1.1481	1.9133
15	73	1.1781	1.8620
16	65	1.2041	1.8139
17	59	1.2304	1.7688
18	53	1.2553	1.7263
19	49	1.2788	1.6860
20	44	1.301	1.6478
21	41	1.3222	1.6115
22	38	1.3424	1.5769
23	35	1.3617	1.5438
24	33	1.3802	1.5121
25	30	1.3878	1.4817

Box size,s	N(s)	Log (s)	Log N(s)
26	28	1.415	1.4525
27	27	1.4314	1.4244
28	25	1.4472	1.3974
29	24	1.4624	1.3712
30	22	1.4771	1.3460
31	21	1.4814	1.3216
32	20	1.5051	1.2980
33	19	1.5185	1.2751
34	18	1.5315	1.2528
35	17	1.5441	1.2313
36	16	1.5563	1.2103
37	15	1.5682	1.1899
38	15	1.5798	1.1701
39	14	1.5811	1.1507
40	14	1.6021	1.1319
41	13	1.6128	1.1135
42	12	1.6232	1.0956
43	12	1.6335	1.0780
44	12	1.6435	1.0609
45	11	1.6532	1.0442
46	11	1.6628	1.0278
47	10	1.6721	1.0118
48	10	1.6812	0.9962
49	10	1.6902	0.9808
50	9	1.699	0.9658

Table 4.6: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (c)

Box size,s	N(s)	Log (s)	Log N(s)
1	9324	0	3.9696
2	2408	0.301	3.3817
3	1131	0.4771	3.0535
4	667	0.6021	2.8241
5	580	0.699	2.7632
8	423	0.7782	2.6265
7	324	0.8451	2.5110
8	258	0.8031	2.4109
8	210	0.9542	2.3226
10	175	1	2.2436
11	149	1.0414	2.1722
12	128	1.0782	2.1069
13	111	1.1138	2.0469
14	98	1.1481	1.9914
15	87	1.1781	1.9397
16	78	1.2041	1.8913
17	70	1.2304	1.8458
18	64	1.2553	1.8030
19	58	1.2788	1.7625
20	53	1.301	1.7240
21	49	1.3222	1.6874
22	45	1.3424	1.6526
23	42	1.3617	1.6193
24	39	1.3802	1.5874
25	36	1.3878	1.5568

Box size,s	N(s)	Log (s)	Log N(s)
26	34	1.415	1.5274
27	32	1.4314	1.4991
28	30	1.4472	1.4718
29	28	1.4624	1.4455
30	26	1.4771	1.4201
31	25	1.4814	1.3955
32	24	1.5051	1.3717
33	22	1.5185	1.3486
34	21	1.5315	1.3263
35	20	1.5441	1.3045
36	19	1.5563	1.2834
37	18	1.5682	1.2629
38	17	1.5798	1.2429
39	17	1.5811	1.2234
40	16	1.6021	1.2044
41	15	1.6128	1.1859
42	15	1.6232	1.1679
43	14	1.6335	1.1502
44	14	1.6435	1.1330
45	13	1.6532	1.1162
46	13	1.6628	1.0997
47	12	1.6721	1.0836
48	12	1.6812	1.0678
49	11	1.6902	1.0523
50	11	1.699	1.0372

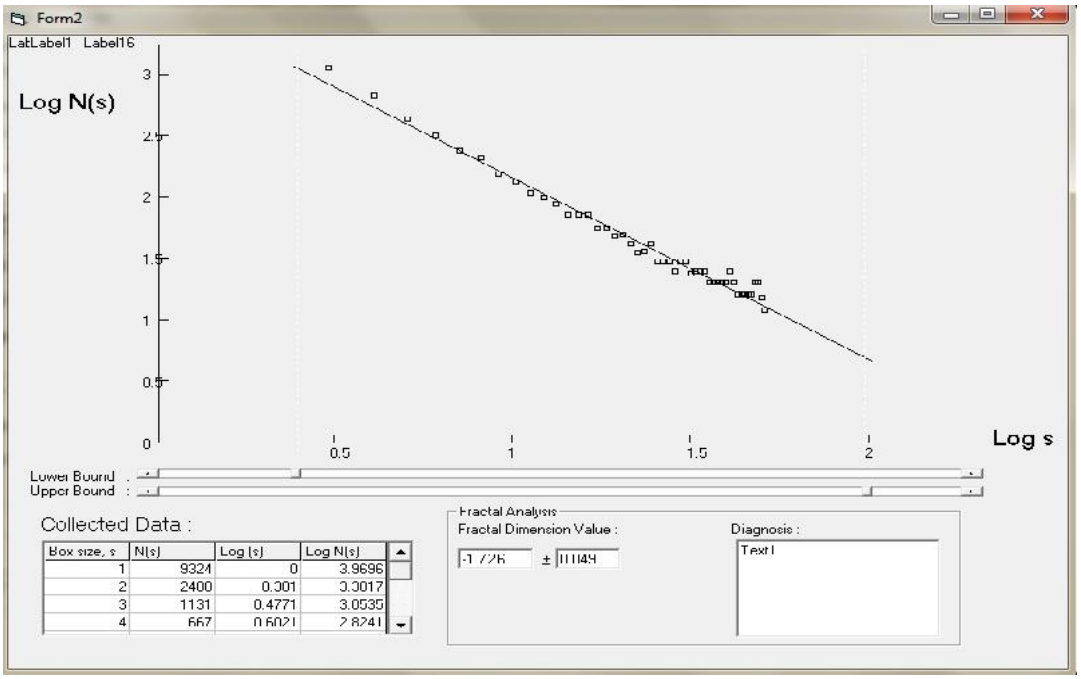


Figure 4.6(c): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.726 for the image of Figure 4.1(c)

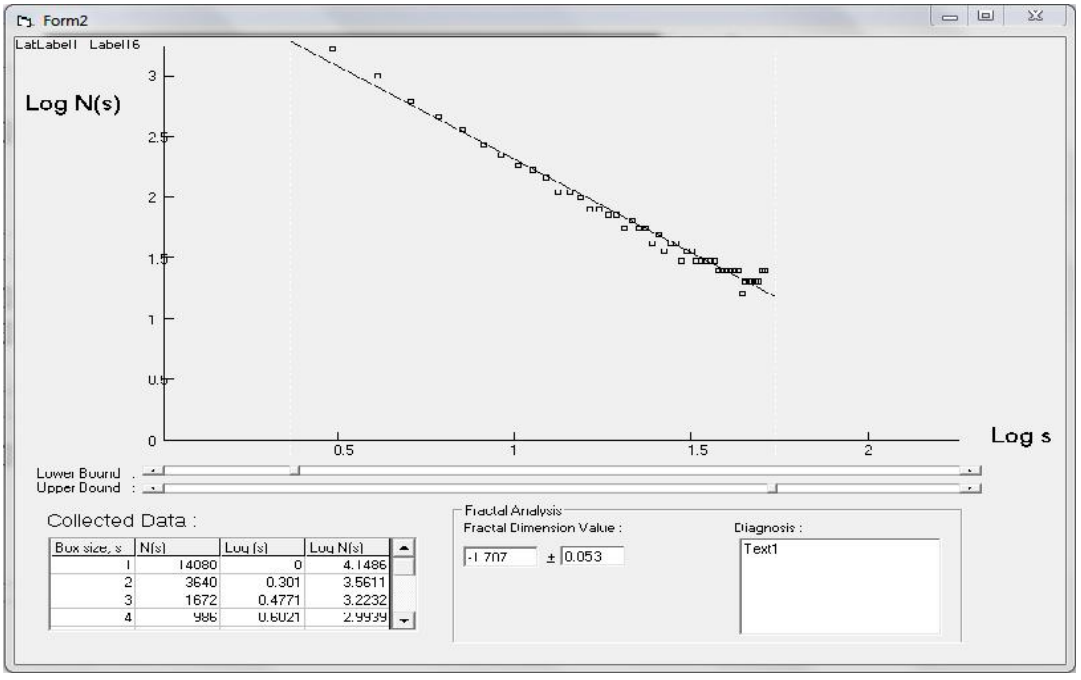


Figure 4.6(d): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.707 for the image of Figure 4.1(d)

Table 4.7: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (d)

Box size,s	N(s)	Log (s)	Log N(s)
1	14080	0	4.1486
2	3640	0.301	3.5611
3	1672	0.4771	3.2232
4	986	0.6021	2.9939
5	903	0.699	2.9555
8	661	0.7782	2.8203
7	508	0.8451	2.7060
8	405	0.8031	2.6070
8	331	0.9542	2.5197
10	276	1	2.4416
11	235	1.0414	2.3709
12	203	1.0782	2.3064
13	177	1.1138	2.2471
14	156	1.1481	2.1922
15	138	1.1781	2.1410
16	124	1.2041	2.0932
17	112	1.2304	2.0482
18	101	1.2553	2.0058
19	92	1.2788	1.9658
20	85	1.301	1.9277
21	78	1.3222	1.8916
22	72	1.3424	1.8571
23	67	1.3617	1.8241
24	62	1.3802	1.7926
25	58	1.3878	1.7623

Box size,s	N(s)	Log (s)	Log N(s)
26	54	1.415	1.7332
27	51	1.4314	1.7053
28	48	1.4472	1.6783
29	45	1.4624	1.6523
30	42	1.4771	1.6272
31	40	1.4814	1.6028
32	38	1.5051	1.5793
33	36	1.5185	1.5565
34	34	1.5315	1.5344
35	33	1.5441	1.5129
36	31	1.5563	1.4920
37	30	1.5682	1.4717
38	28	1.5798	1.4519
39	27	1.5811	1.4327
40	26	1.6021	1.4139
41	25	1.6128	1.3956
42	24	1.6232	1.3777
43	23	1.6335	1.3603
44	22	1.6435	1.3432
45	21	1.6532	1.3266
46	20	1.6628	1.3103
47	20	1.6721	1.2943
48	19	1.6812	1.2787
49	18	1.6902	1.2634
50	18	1.699	1.2485

Table 4.8: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (e)

Box size,s	N(s)	Log (s)	Log N(s)
1	11328	0	4.0542
2	2940	0.301	3.4683
3	1320	0.4771	3.1206
4	806	0.6021	2.9063
5	737	0.699	2.8673
8	541	0.7782	2.7329
7	416	0.8451	2.6192
8	332	0.8031	2.5208
8	272	0.9542	2.4339
10	227	1	2.3562
11	193	1.0414	2.2859
12	167	1.0782	2.2218
13	145	1.1138	2.1627
14	128	1.1481	2.1081
15	114	1.1781	2.0572
16	102	1.2041	2.0096
17	92	1.2304	1.9649
18	84	1.2553	1.9227
19	76	1.2788	1.8829
20	70	1.301	1.8451
21	64	1.3222	1.8091
22	60	1.3424	1.7748
23	55	1.3617	1.7420
24	51	1.3802	1.7106
25	48	1.3878	1.6805

Box size,s	N(s)	Log (s)	Log N(s)
26	45	1.415	1.6516
27	42	1.4314	1.6237
28	40	1.4472	1.5969
29	37	1.4624	1.5710
30	35	1.4771	1.5460
31	33	1.4814	1.5219
32	32	1.5051	1.4985
33	30	1.5185	1.4758
34	28	1.5315	1.4537
35	27	1.5441	1.4324
36	26	1.5563	1.4116
37	25	1.5682	1.3914
38	24	1.5798	1.3717
39	23	1.5811	1.3526
40	22	1.6021	1.3339
41	21	1.6128	1.3157
42	20	1.6232	1.2979
43	19	1.6335	1.2806
44	18	1.6435	1.2636
45	18	1.6532	1.2470
46	17	1.6628	1.2308
47	16	1.6721	1.2150
48	16	1.6812	1.1995
49	15	1.6902	1.1842
50	15	1.699	1.1693

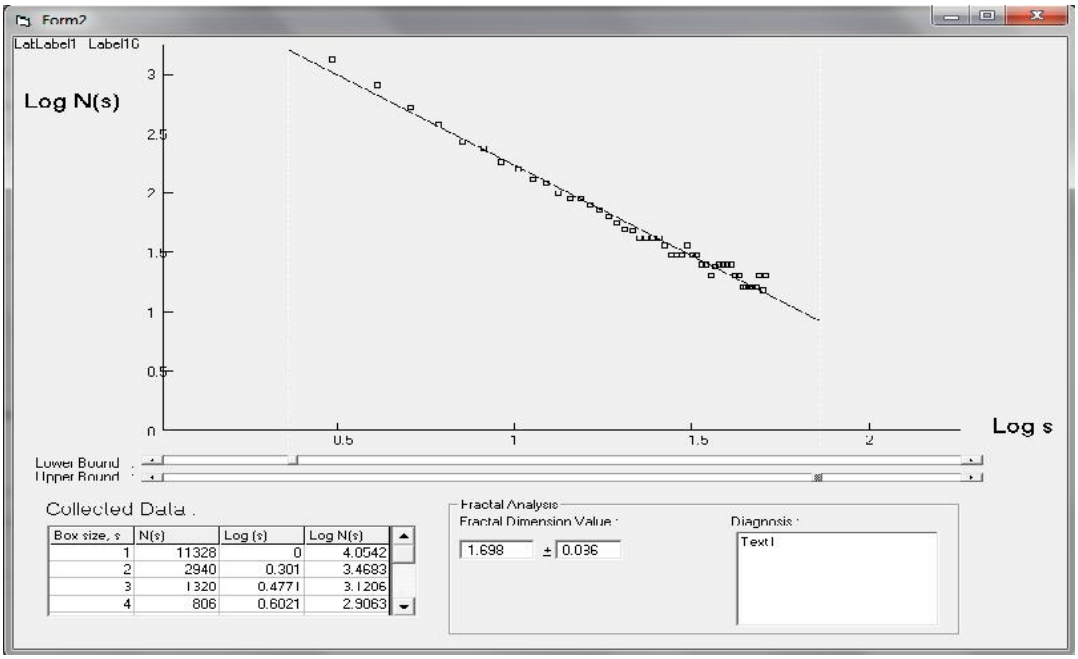


Figure 4.6(e): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.698 for the image of Figure 4.1(e)

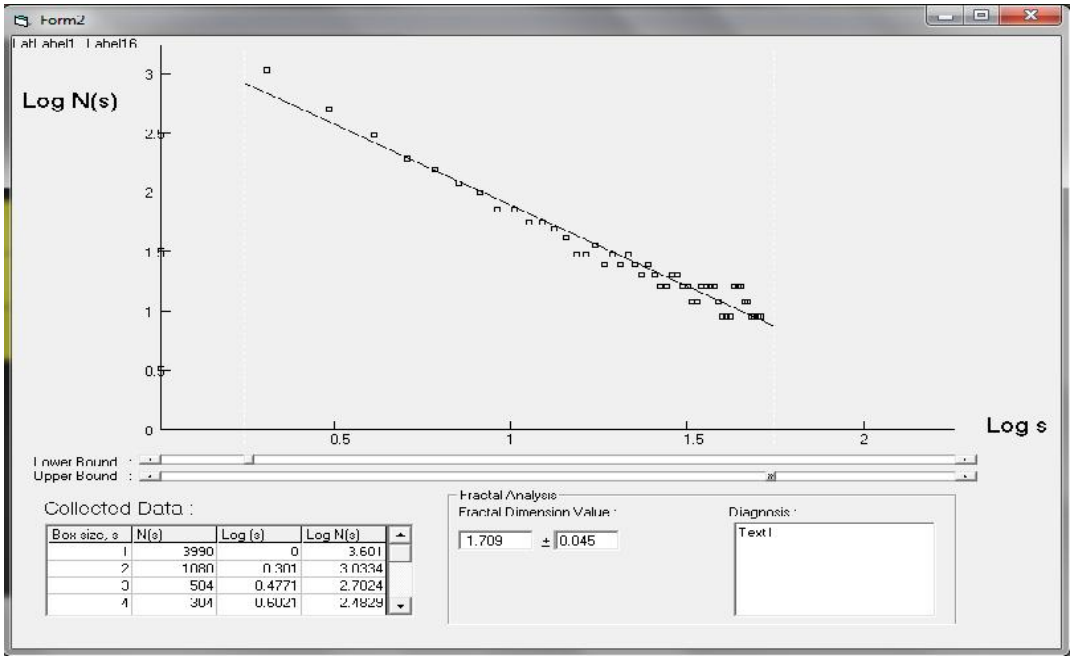


Figure 4.6(f): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.709 for the image of Figure 4.1(f)

Table 4.9: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (f)

Box size,s	N(s)	Log (s)	Log N(s)
1	3990	0	3.6010
2	1080	0.301	3.0334
3	504	0.4771	2.7024
4	304	0.6021	2.4829
5	378	0.699	2.5778
8	277	0.7782	2.4424
7	213	0.8451	2.3280
8	169	0.8031	2.2289
8	139	0.9542	2.1415
10	116	1	2.0633
11	98	1.0414	1.9926
12	85	1.0782	1.9280
13	74	1.1138	1.8686
14	65	1.1481	1.8136
15	58	1.1781	1.7624
16	52	1.2041	1.7145
17	47	1.2304	1.6695
18	42	1.2553	1.6270
19	39	1.2788	1.5869
20	35	1.301	1.5488
21	33	1.3222	1.5126
22	30	1.3424	1.4781
23	28	1.3617	1.4451
24	26	1.3802	1.4135
25	24	1.3878	1.3832

Box size,s	N(s)	Log (s)	Log N(s)
26	25	1.415	1.1828
27	25	1.4314	1.1548
28	25	1.4472	1.1278
29	25	1.4624	1.1018
30	25	1.4771	1.0766
31	25	1.4814	1.0523
32	20	1.5051	1.0287
33	16	1.5185	1.0059
34	16	1.5315	0.9837
35	20	1.5441	0.9622
36	20	1.5563	0.9413
37	16	1.5682	0.9209
38	16	1.5798	0.9011
39	16	1.5811	0.8819
40	16	1.6021	0.8631
41	16	1.6128	0.8448
42	12	1.6232	0.8269
43	9	1.6335	0.8094
44	9	1.6435	0.7923
45	9	1.6532	0.7757
46	12	1.6628	0.7593
47	12	1.6721	0.7434
48	16	1.6812	0.7278
49	16	1.6902	0.7125
50	16	1.699	0.6975

Table 4.10: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (g)

Box size,s	N(s)	Log (s)	Log N(s)
1	19680	0	4.2940
2	5022	0.301	3.7009
3	2365	0.4771	3.3738
4	1312	0.6021	3.1179
5	1214	0.699	3.0841
8	885	0.7782	2.9470
7	678	0.8451	2.8311
8	538	0.8031	2.7308
8	439	0.9542	2.6422
10	366	1	2.5630
11	310	1.0414	2.4913
12	267	1.0782	2.4259
13	232	1.1138	2.3658
14	204	1.1481	2.3101
15	181	1.1781	2.2582
16	162	1.2041	2.2097
17	146	1.2304	2.1641
18	132	1.2553	2.1211
19	120	1.2788	2.0805
20	110	1.301	2.0419
21	101	1.3222	2.0052
22	93	1.3424	1.9703
23	86	1.3617	1.9368
24	80	1.3802	1.9049
25	75	1.3878	1.8742

Box size,s	N(s)	Log (s)	Log N(s)
26	70	1.415	1.8447
27	66	1.4314	1.8163
28	62	1.4472	1.7890
29	58	1.4624	1.7626
30	55	1.4771	1.7371
31	52	1.4814	1.7125
32	49	1.5051	1.6886
33	46	1.5185	1.6655
34	44	1.5315	1.6430
35	42	1.5441	1.6212
36	40	1.5563	1.6000
37	38	1.5682	1.5794
38	36	1.5798	1.5594
39	35	1.5811	1.5399
40	33	1.6021	1.5208
41	32	1.6128	1.5023
42	30	1.6232	1.4842
43	29	1.6335	1.4665
44	28	1.6435	1.4492
45	27	1.6532	1.4323
46	26	1.6628	1.4158
47	25	1.6721	1.3996
48	24	1.6812	1.3838
49	23	1.6902	1.3683
50	23	1.699	1.3531

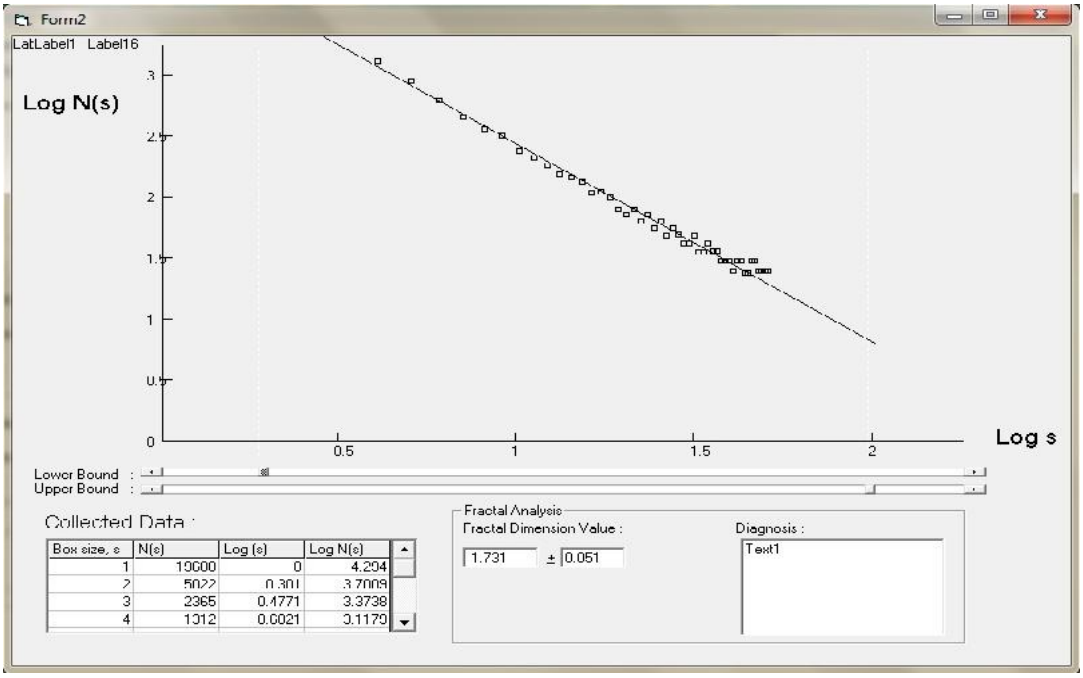


Figure 4.6(g): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.731 for the image of Figure 4.1(g)

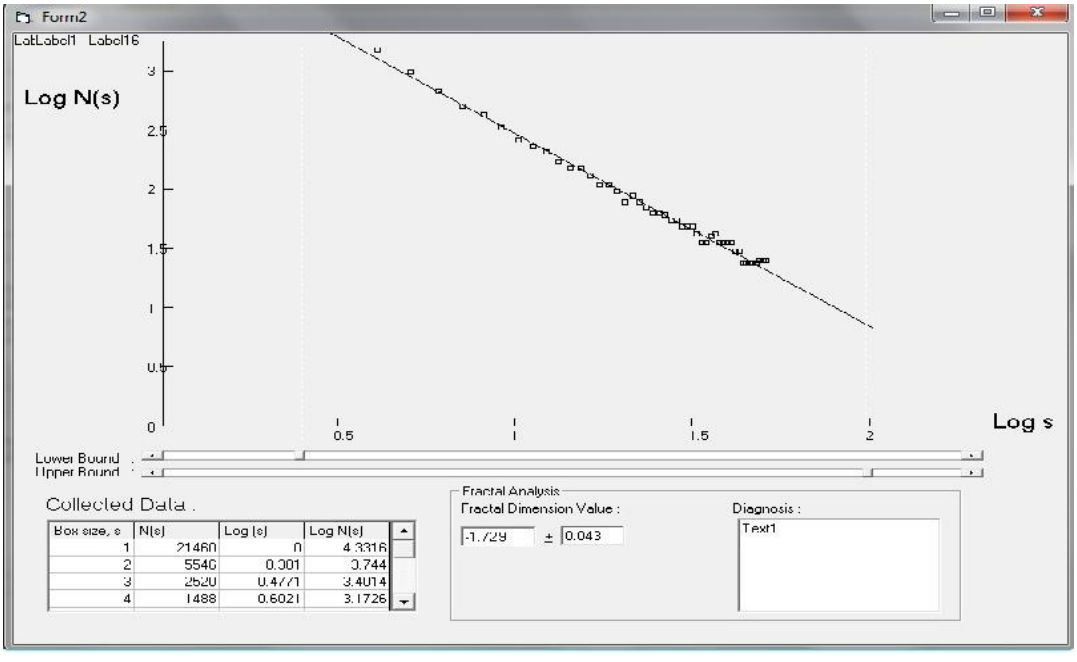


Figure 4.6(h): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.729 for the image of Figure 4.1(h)

Table 4.11: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (h)

Box size,s	N(s)	Log (s)	Log N(s)
1	21460	0	4.3316
2	5546	0.301	3.7440
3	2520	0.4771	3.4014
4	1488	0.6021	3.1726
5	1328	0.699	3.1231
8	969	0.7782	2.9862
7	742	0.8451	2.8704
8	589	0.8031	2.7702
8	481	0.9542	2.6817
10	400	1	2.6026
11	340	1.0414	2.5310
12	292	1.0782	2.4657
13	254	1.1138	2.4056
14	224	1.1481	2.3499
15	199	1.1781	2.2981
16	178	1.2041	2.2497
17	160	1.2304	2.2042
18	145	1.2553	2.1612
19	132	1.2788	2.1206
20	121	1.301	2.0821
21	111	1.3222	2.0455
22	102	1.3424	2.0106
23	95	1.3617	1.9772
24	88	1.3802	1.9452
25	82	1.3878	1.9146

Box size,s	N(s)	Log (s)	Log N(s)
26	77	1.415	1.8851
27	72	1.4314	1.8568
28	68	1.4472	1.8295
29	64	1.4624	1.8031
30	60	1.4771	1.7777
31	57	1.4814	1.7530
32	54	1.5051	1.7292
33	51	1.5185	1.7061
34	48	1.5315	1.6837
35	46	1.5441	1.6619
36	44	1.5563	1.6408
37	42	1.5682	1.6202
38	40	1.5798	1.6002
39	38	1.5811	1.5806
40	36	1.6021	1.5616
41	35	1.6128	1.5431
42	33	1.6232	1.5250
43	32	1.6335	1.5073
44	31	1.6435	1.4901
45	30	1.6532	1.4732
46	29	1.6628	1.4567
47	28	1.6721	1.4405
48	27	1.6812	1.4247
49	26	1.6902	1.4093
50	25	1.699	1.3941

Table 4.12: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (i)

Box size,s	N(s)	Log (s)	Log N(s)
1	37830	0	4.5778
2	9604	0.301	3.9825
3	4422	0.4771	3.6456
4	2500	0.6021	3.3979
5	2177	0.699	3.3378
8	1681	0.7782	3.2256
7	1575	0.8451	3.1974
8	1198	0.8031	3.0786
8	946	0.9542	2.9757
10	767	1	2.8850
11	637	1.0414	2.8038
12	537	1.0782	2.7304
13	461	1.1138	2.6633
14	400	1.1481	2.6017
15	350	1.1781	2.5446
16	310	1.2041	2.4914
17	276	1.2304	2.4417
18	248	1.2553	2.3950
19	224	1.2788	2.3509
20	186	1.301	2.2698
21	171	1.3222	2.2322
22	157	1.3424	2.1963
23	145	1.3617	2.1621
24	135	1.3802	2.1293
25	125	1.3878	2.0979

Box size,s	N(s)	Log (s)	Log N(s)
26	117	1.415	2.0676
27	109	1.4314	2.0386
28	102	1.4472	2.0105
29	96	1.4624	1.9835
30	91	1.4771	1.9574
31	86	1.4814	1.9321
32	81	1.5051	1.9077
33	77	1.5185	1.8840
34	73	1.5315	1.8610
35	69	1.5441	1.8386
36	66	1.5563	1.8169
37	62	1.5682	1.7958
38	60	1.5798	1.7753
39	57	1.5811	1.7553
40	54	1.6021	1.7357
41	52	1.6128	1.7167
42	50	1.6232	1.6982
43	48	1.6335	1.6800
44	46	1.6435	1.6623
45	44	1.6532	1.6450
46	42	1.6628	1.6281
47	41	1.6721	1.6115
48	39	1.6812	1.5953
49	38	1.6902	1.5794
50	37	1.699	1.5638

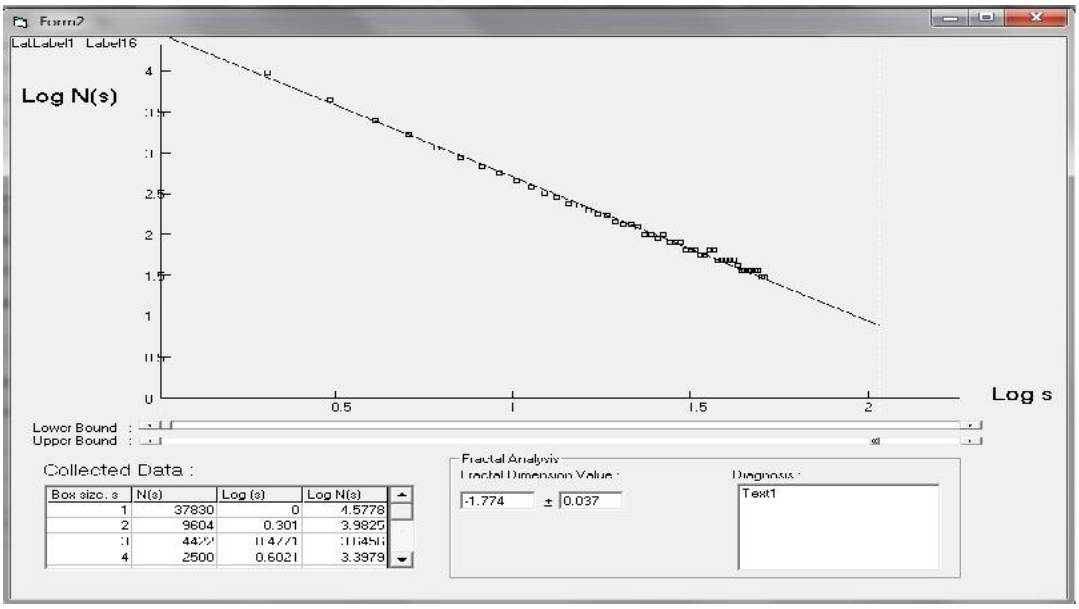


Figure 4.6(i): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.774 for the image of Figure 4.1(i)

The data for box count dimension of the images of Figure 4.2(a) to (i) are tabulated in Table 4.13 to 4.21 with their screen shots of the graphs which yield their fractal dimension values in Figure 4.7(a) to (i) respectively.

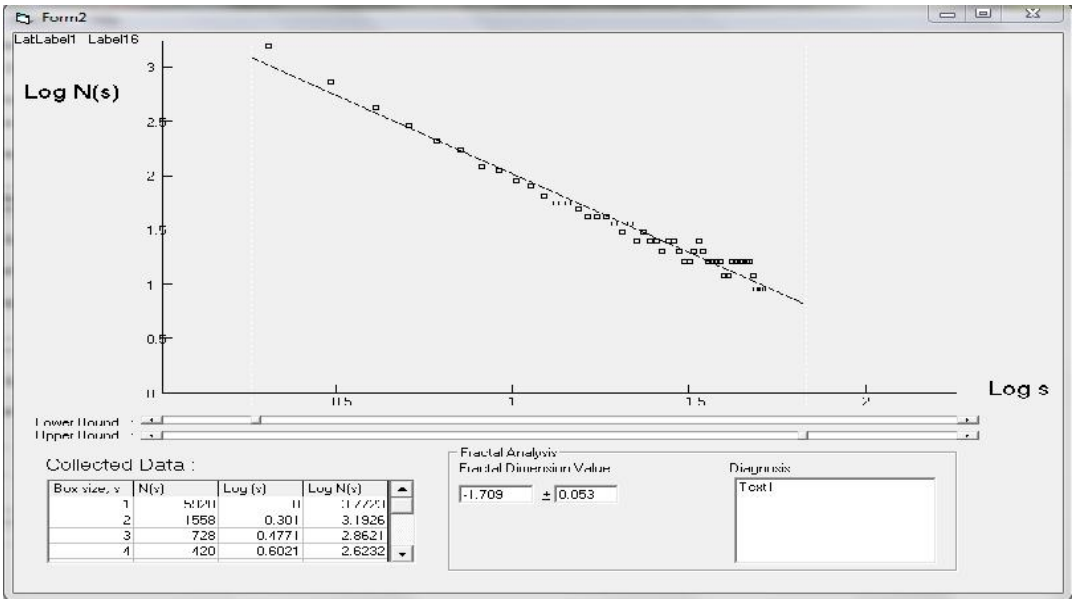


Figure 4.7(a): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.709 for the image of Figure 4.2(a)

Table 4.13: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (a)

Box size,s	N(s)	Log (s)	Log N(s)
1	5920	0	3.7723
2	1558	0.301	3.1926
3	728	0.4771	2.8621
4	420	0.6021	2.6232
5	378	0.699	2.5778
8	277	0.7782	2.4424
7	213	0.8451	2.3280
8	169	0.8031	2.2289
9	139	0.9542	2.1415
10	116	1	2.0633
11	98	1.0414	1.9926
12	85	1.0782	1.9280
13	74	1.1138	1.8686
14	65	1.1481	1.8136
15	58	1.1781	1.7624
16	52	1.2041	1.7145
17	47	1.2304	1.6695
18	42	1.2553	1.6270
19	39	1.2788	1.5869
20	35	1.301	1.5488
21	33	1.3222	1.5126
22	30	1.3424	1.4781
23	28	1.3617	1.4451
24	26	1.3802	1.4135
25	24	1.3878	1.3832

Box size,s	N(s)	Log (s)	Log N(s)
26	23	1.415	1.3541
27	21	1.4314	1.3261
28	20	1.4472	1.2991
29	19	1.4624	1.2731
30	18	1.4771	1.2479
31	17	1.4814	1.2236
32	16	1.5051	1.2000
33	15	1.5185	1.1772
34	14	1.5315	1.1550
35	14	1.5441	1.1335
36	13	1.5563	1.1126
37	12	1.5682	1.0922
38	12	1.5798	1.0724
39	11	1.5811	1.0532
40	11	1.6021	1.0344
41	10	1.6128	1.0161
42	10	1.6232	0.9982
43	10	1.6335	0.9807
44	9	1.6435	0.9636
45	9	1.6532	0.9470
46	9	1.6628	0.9306
47	8	1.6721	0.9147
48	8	1.6812	0.8991
49	8	1.6902	0.8838
50	7	1.699	0.8688

Table 4.14: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (b)

Box size,s	N(s)	Log (s)	Log N(s)
1	6160	0	3.7896
2	1653	0.301	3.2694
3	780	0.4771	2.9651
4	450	0.6021	2.7492
5	382	0.699	2.5818
8	279	0.7782	2.4450
7	213	0.8451	2.3293
8	169	0.8031	2.2291
9	138	0.9542	2.1407
10	115	1	2.0616
11	98	1.0414	1.9901
12	84	1.0782	1.9248
13	73	1.1138	1.8647
14	64	1.1481	1.8091
15	57	1.1781	1.7573
16	51	1.2041	1.7089
17	46	1.2304	1.6634
18	42	1.2553	1.6205
19	38	1.2788	1.5799
20	35	1.301	1.5414
21	32	1.3222	1.5048
22	30	1.3424	1.4699
23	27	1.3617	1.4365
24	25	1.3802	1.4046
25	24	1.3878	1.3740

Box size,s	N(s)	Log (s)	Log N(s)
26	22	1.415	1.3445
27	21	1.4314	1.3162
28	19	1.4472	1.2889
29	18	1.4624	1.2626
30	17	1.4771	1.2371
31	16	1.4814	1.2125
32	15	1.5051	1.1887
33	15	1.5185	1.1656
34	14	1.5315	1.1432
35	13	1.5441	1.1215
36	13	1.5563	1.1003
37	12	1.5682	1.0797
38	11	1.5798	1.0597
39	11	1.5811	1.0402
40	11	1.6021	1.0212
41	10	1.6128	1.0027
42	10	1.6232	0.9846
43	9	1.6335	0.9670
44	9	1.6435	0.9497
45	9	1.6532	0.9328
46	8	1.6628	0.9164
47	8	1.6721	0.9002
48	8	1.6812	0.8844
49	7	1.6902	0.8689
50	7	1.699	0.8538

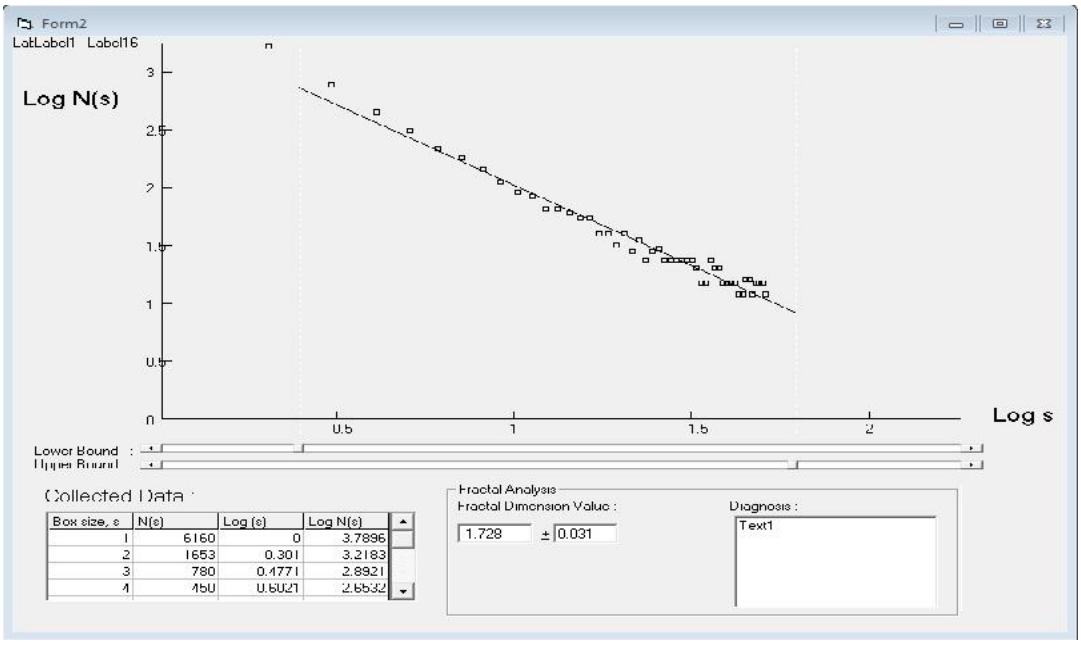


Figure 4.7(b): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.728 for the image of Figure 4.2(b)

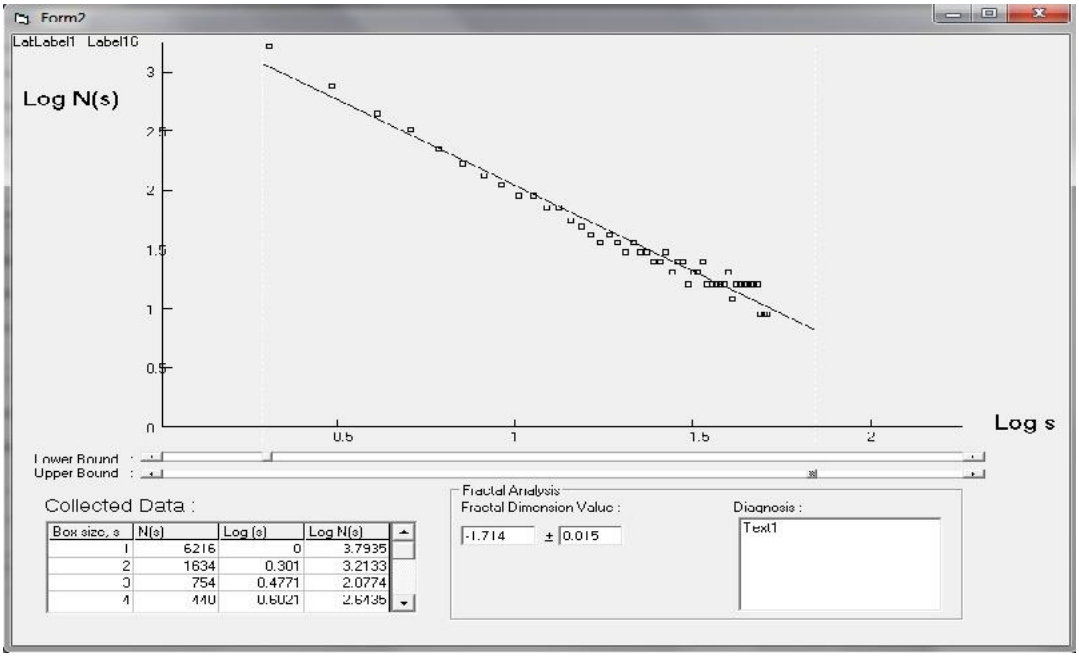


Figure 4.7(c): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.714 for the image of Figure 4.2(c)

Table 4.15: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (c)

Box size,s	N(s)	Log (s)	Log N(s)
1	6216	0	3.7935
2	1634	0.301	3.2133
3	754	0.4771	2.8774
4	440	0.6021	2.6435
5	323	0.699	2.5092
8	224	0.7782	2.3502
7	221	0.8451	2.3450
8	176	0.8031	2.2456
8	144	0.9542	2.1579
10	120	1	2.0795
11	102	1.0414	2.0086
12	88	1.0782	1.9438
13	77	1.1138	1.8842
14	67	1.1481	1.8290
15	60	1.1781	1.7777
16	54	1.2041	1.7296
17	48	1.2304	1.6845
18	44	1.2553	1.6420
19	40	1.2788	1.6017
20	37	1.301	1.5635
21	34	1.3222	1.5272
22	31	1.3424	1.4926
23	29	1.3617	1.4595
24	27	1.3802	1.4278
25	25	1.3878	1.3974

Box size,s	N(s)	Log (s)	Log N(s)
26	23	1.415	1.3682
27	22	1.4314	1.3401
28	21	1.4472	1.3131
29	19	1.4624	1.2869
30	18	1.4771	1.2617
31	17	1.4814	1.2373
32	16	1.5051	1.2137
33	16	1.5185	1.1908
34	15	1.5315	1.1685
35	14	1.5441	1.1470
36	13	1.5563	1.1260
37	13	1.5682	1.1056
38	12	1.5798	1.0858
39	12	1.5811	1.0664
40	11	1.6021	1.0476
41	11	1.6128	1.0292
42	10	1.6232	1.0113
43	10	1.6335	0.9937
44	9	1.6435	0.9766
45	9	1.6532	0.9599
46	9	1.6628	0.9435
47	8	1.6721	0.9275
48	8	1.6812	0.9119
49	8	1.6902	0.8965
50	8	1.699	0.8815

Table 4.16: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (d)

Box size,s	N(s)	Log (s)	Log N(s)
1	8820	0	3.9455
2	2322	0.301	3.3659
3	1073	0.4771	3.0306
4	644	0.6021	2.8089
5	526	0.699	2.7208
8	381	0.7782	2.5808
7	290	0.8451	2.4625
8	229	0.8031	2.3599
8	186	0.9542	2.2695
10	154	1	2.1886
11	130	1.0414	2.1154
12	112	1.0782	2.0486
13	97	1.1138	1.9871
14	85	1.1481	1.9302
15	75	1.1781	1.8773
16	67	1.2041	1.8277
17	60	1.2304	1.7812
18	55	1.2553	1.7373
19	50	1.2788	1.6958
20	45	1.301	1.6564
21	42	1.3222	1.6189
22	38	1.3424	1.5832
23	35	1.3617	1.5491
24	33	1.3802	1.5164
25	31	1.3878	1.4850
26	29	1.415	1.4651
27	27	1.4314	1.4363
28	26	1.4472	1.4086
29	24	1.4624	1.3819
30	23	1.4771	1.3561
31	21	1.4814	1.3311
32	20	1.5051	1.3070
33	19	1.5185	1.2835
34	18	1.5315	1.2608
35	17	1.5441	1.2387
36	16	1.5563	1.2173
37	16	1.5682	1.1964
38	15	1.5798	1.1761
39	14	1.5811	1.1564
40	14	1.6021	1.1371
41	13	1.6128	1.1183
42	13	1.6232	1.0999
43	12	1.6335	1.0820
44	12	1.6435	1.0645
45	11	1.6532	1.0474
46	11	1.6628	1.0307
47	10	1.6721	1.0143
48	10	1.6812	0.9983
49	10	1.6902	0.9826
50	9	1.699	0.9672

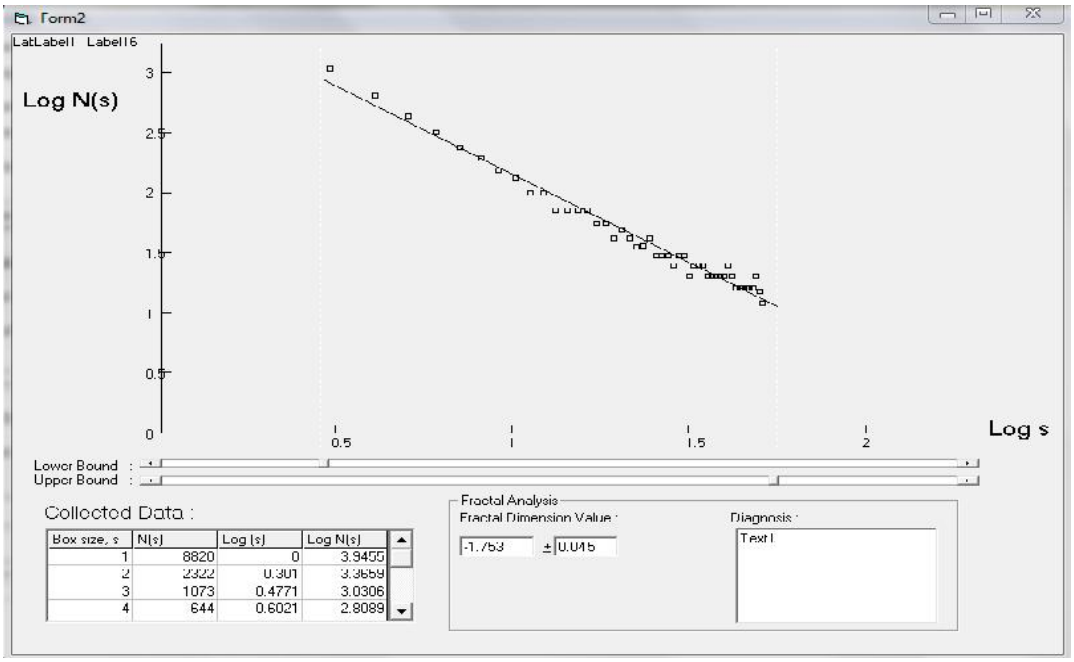


Figure 4.7(d): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.753 for the image of Figure 4.2(d)

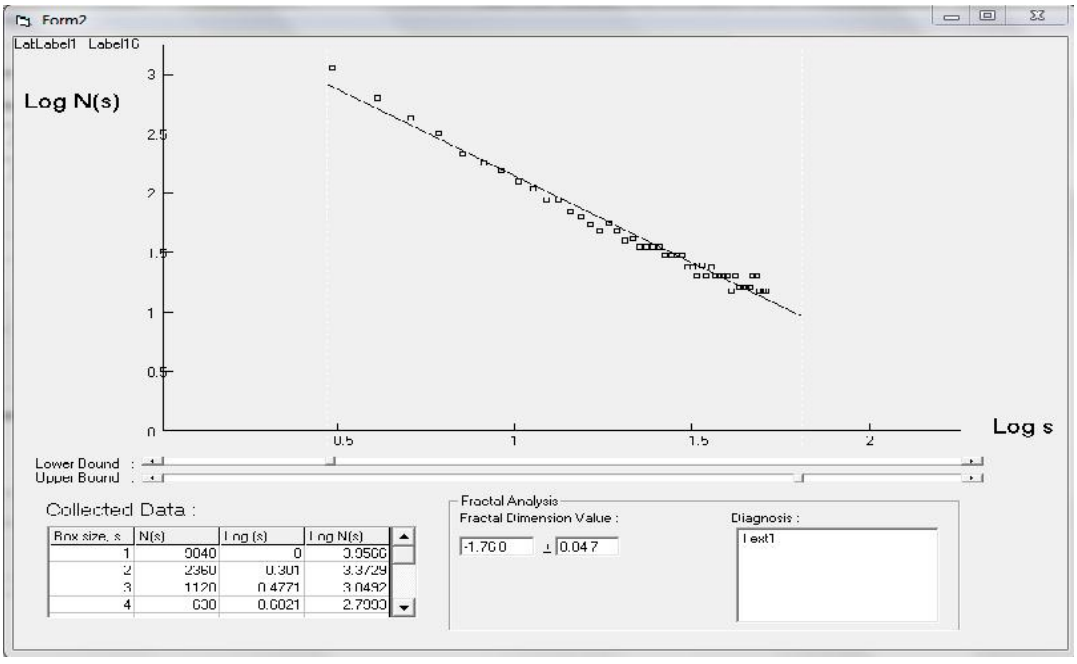


Figure 4.7(e): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.768 for the image of Figure 4.2(e)

Table 4.17: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (e)

Box size,s	N(s)	Log (s)	Log N(s)
1	9048	0	3.9566
2	2360	0.301	3.3729
3	1120	0.4771	3.0492
4	630	0.6021	2.7993
5	526	0.699	2.7208
8	381	0.7782	2.5808
7	290	0.8451	2.4625
8	229	0.8031	2.3599
8	186	0.9542	2.2695
10	154	1	2.1886
11	130	1.0414	2.1154
12	112	1.0782	2.0486
13	97	1.1138	1.9871
14	85	1.1481	1.9302
15	75	1.1781	1.8773
16	67	1.2041	1.8277
17	60	1.2304	1.7812
18	55	1.2553	1.7373
19	50	1.2788	1.6958
20	45	1.301	1.6564
21	42	1.3222	1.6189
22	38	1.3424	1.5832
23	35	1.3617	1.5491
24	33	1.3802	1.5164
25	31	1.3878	1.4850
26	29	1.415	1.4549
27	27	1.4314	1.4259
28	25	1.4472	1.3980
29	24	1.4624	1.3711
30	22	1.4771	1.3450
31	21	1.4814	1.3199
32	20	1.5051	1.2955
33	19	1.5185	1.2719
34	18	1.5315	1.2489
35	17	1.5441	1.2267
36	16	1.5563	1.2051
37	15	1.5682	1.1840
38	15	1.5798	1.1635
39	14	1.5811	1.1436
40	13	1.6021	1.1242
41	13	1.6128	1.1052
42	12	1.6232	1.0867
43	12	1.6335	1.0686
44	11	1.6435	1.0510
45	11	1.6532	1.0337
46	10	1.6628	1.0168
47	10	1.6721	1.0003
48	10	1.6812	0.9842
49	9	1.6902	0.9683
50	9	1.699	0.9528

Table 4.18: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.1 (f)

Box size,s	N(s)	Log (s)	Log N(s)
1	14157	0	4.1510
2	3660	0.301	3.5635
3	1722	0.4771	3.2360
4	961	0.6021	2.9827
5	859	0.699	2.9341
8	626	0.7782	2.7962
7	478	0.8451	2.6797
8	379	0.8031	2.5787
8	309	0.9542	2.4897
10	257	1	2.4100
11	218	1.0414	2.3379
12	187	1.0782	2.2721
13	163	1.1138	2.2116
14	143	1.1481	2.1556
15	127	1.1781	2.1034
16	113	1.2041	2.0546
17	102	1.2304	2.0088
18	92	1.2553	1.9656
19	84	1.2788	1.9247
20	77	1.301	1.8859
21	71	1.3222	1.8490
22	65	1.3424	1.8138
23	60	1.3617	1.7802
24	56	1.3802	1.7481
25	52	1.3878	1.7172

Box size,s	N(s)	Log (s)	Log N(s)
26	49	1.415	1.6875
27	46	1.4314	1.6590
28	43	1.4472	1.6315
29	40	1.4624	1.6050
30	38	1.4771	1.5793
31	36	1.4814	1.5545
32	34	1.5051	1.5305
33	32	1.5185	1.5073
34	31	1.5315	1.4847
35	29	1.5441	1.4628
36	28	1.5563	1.4415
37	26	1.5682	1.4208
38	25	1.5798	1.4006
39	24	1.5811	1.3810
40	23	1.6021	1.3618
41	22	1.6128	1.3431
42	21	1.6232	1.3249
43	20	1.6335	1.3071
44	19	1.6435	1.2897
45	19	1.6532	1.2728
46	18	1.6628	1.2561
47	17	1.6721	1.2399
48	17	1.6812	1.2240
49	16	1.6902	1.2084
50	16	1.699	1.1931

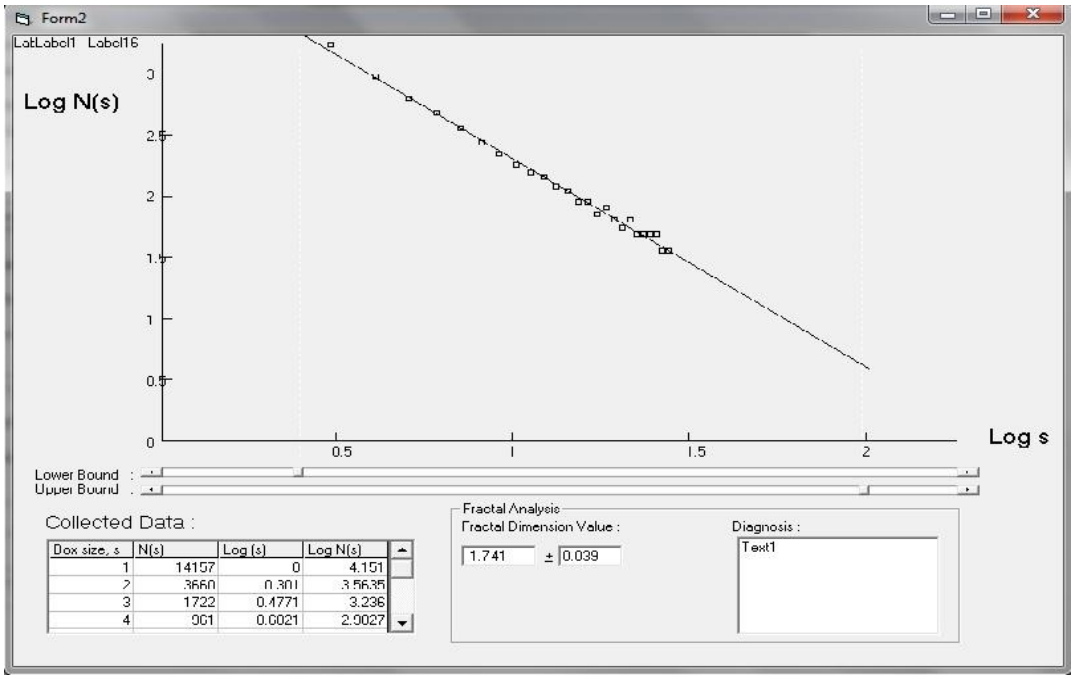


Figure 4.7(f): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.741 for the image of Figure 4.2(f)

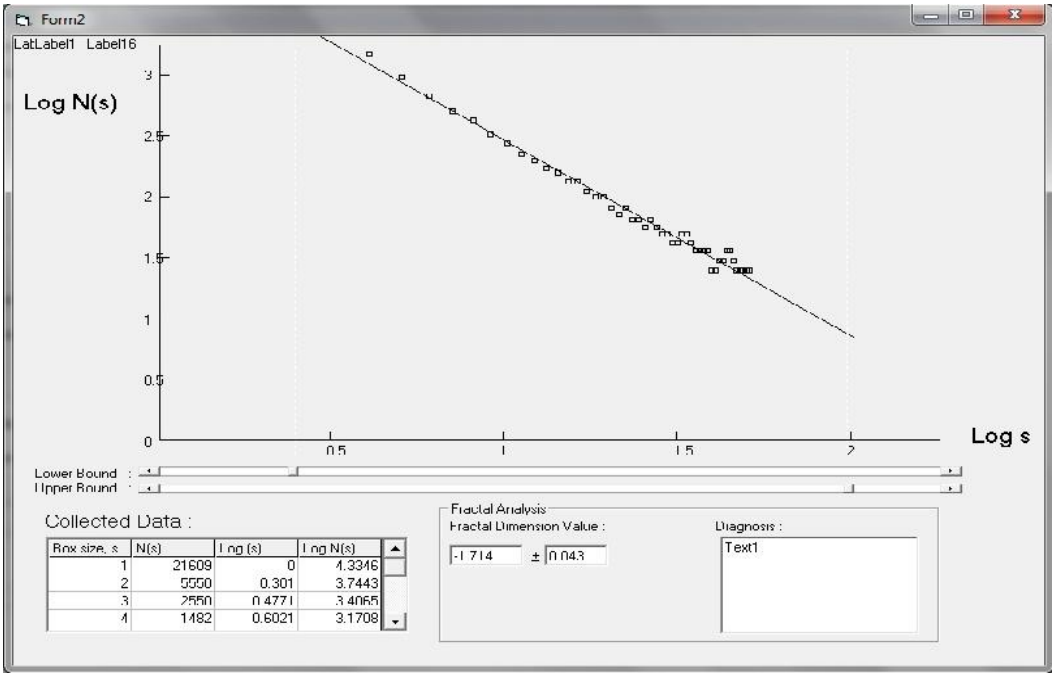


Figure 4.7(g): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.714 for the image of Figure 4.2(g)

Table 4.19: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (g)

Box size,s	N(s)	Log (s)	Log N(s)
1	21609	0	4.3346
2	5550	0.301	3.7443
3	2550	0.4771	3.4065
4	1482	0.6021	3.1708
5	1370	0.699	3.1366
8	1002	0.7782	3.0008
7	769	0.8451	2.8861
8	612	0.8031	2.7867
8	500	0.9542	2.6990
10	417	1	2.6206
11	355	1.0414	2.5497
12	305	1.0782	2.4849
13	266	1.1138	2.4253
14	234	1.1481	2.3701
15	208	1.1781	2.3188
16	187	1.2041	2.2707
17	168	1.2304	2.2256
18	152	1.2553	2.1831
19	139	1.2788	2.1428
20	127	1.301	2.1046
21	117	1.3222	2.0683
22	108	1.3424	2.0337
23	100	1.3617	2.0006
24	93	1.3802	1.9689
25	87	1.3878	1.9385

Box size,s	N(s)	Log (s)	Log N(s)
26	81	1.415	1.9093
27	76	1.4314	1.8812
28	71	1.4472	1.8542
29	67	1.4624	1.8280
30	64	1.4771	1.8028
31	60	1.4814	1.7784
32	57	1.5051	1.7548
33	54	1.5185	1.7319
34	51	1.5315	1.7096
35	49	1.5441	1.6881
36	46	1.5563	1.6671
37	44	1.5682	1.6467
38	42	1.5798	1.6269
39	41	1.5811	1.6075
40	39	1.6021	1.5887
41	37	1.6128	1.5703
42	36	1.6232	1.5524
43	34	1.6335	1.5348
44	33	1.6435	1.5177
45	32	1.6532	1.5010
46	31	1.6628	1.4846
47	29	1.6721	1.4686
48	28	1.6812	1.4530
49	27	1.6902	1.4376
50	26	1.699	1.4226

Table 4.20: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (h)

Box size,s	N(s)	Log (s)	Log N(s)
1	21780	0	4.3381
2	5551	0.301	3.7444
3	2562	0.4771	3.4086
4	1426	0.6021	3.1541
5	1311	0.699	3.1177
8	954	0.7782	2.9794
7	729	0.8451	2.8626
8	577	0.8031	2.7613
8	470	0.9542	2.6720
10	391	1	2.5921
11	331	1.0414	2.5198
12	284	1.0782	2.4538
13	247	1.1138	2.3932
14	217	1.1481	2.3370
15	193	1.1781	2.2846
16	172	1.2041	2.2357
17	155	1.2304	2.1897
18	140	1.2553	2.1464
19	127	1.2788	2.1054
20	117	1.301	2.0665
21	107	1.3222	2.0295
22	99	1.3424	1.9942
23	91	1.3617	1.9605
24	85	1.3802	1.9283
25	79	1.3878	1.8973

Box size,s	N(s)	Log (s)	Log N(s)
26	74	1.415	1.8676
27	69	1.4314	1.8389
28	65	1.4472	1.8114
29	61	1.4624	1.7848
30	57	1.4771	1.7590
31	54	1.4814	1.7342
32	51	1.5051	1.7101
33	49	1.5185	1.6868
34	46	1.5315	1.6641
35	44	1.5441	1.6422
36	42	1.5563	1.6208
37	40	1.5682	1.6000
38	38	1.5798	1.5798
39	36	1.5811	1.5601
40	35	1.6021	1.5409
41	33	1.6128	1.5222
42	32	1.6232	1.5039
43	31	1.6335	1.4861
44	29	1.6435	1.4686
45	28	1.6532	1.4516
46	27	1.6628	1.4349
47	26	1.6721	1.4186
48	25	1.6812	1.4027
49	24	1.6902	1.3870
50	24	1.699	1.3717

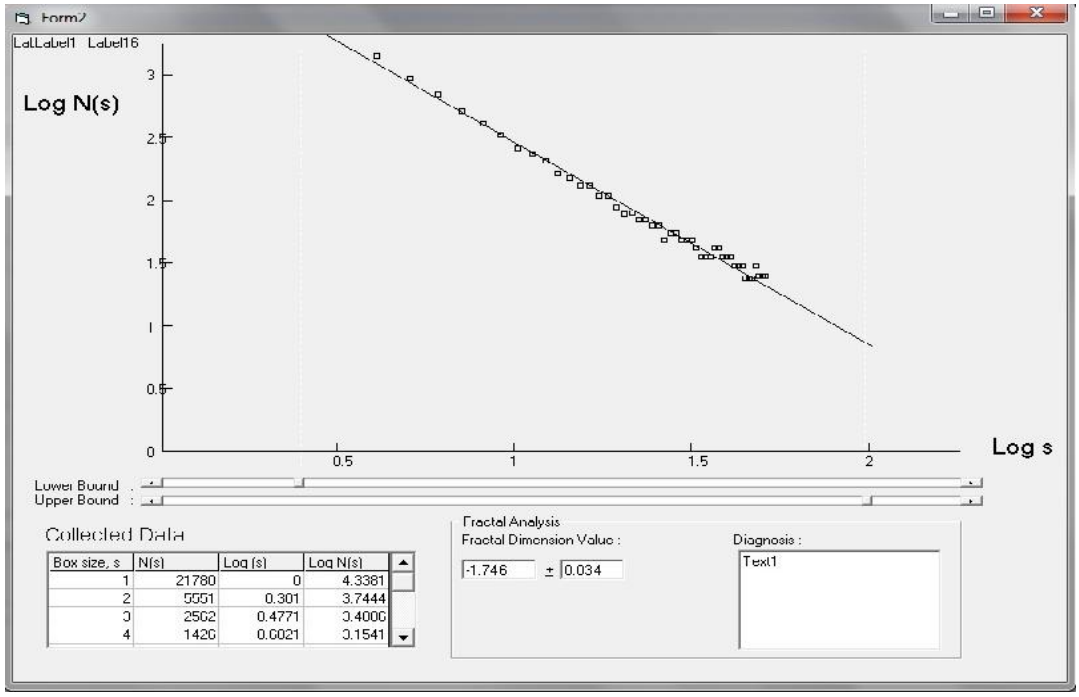


Figure 4.7(h): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.746 for the image of Figure 4.2(h)

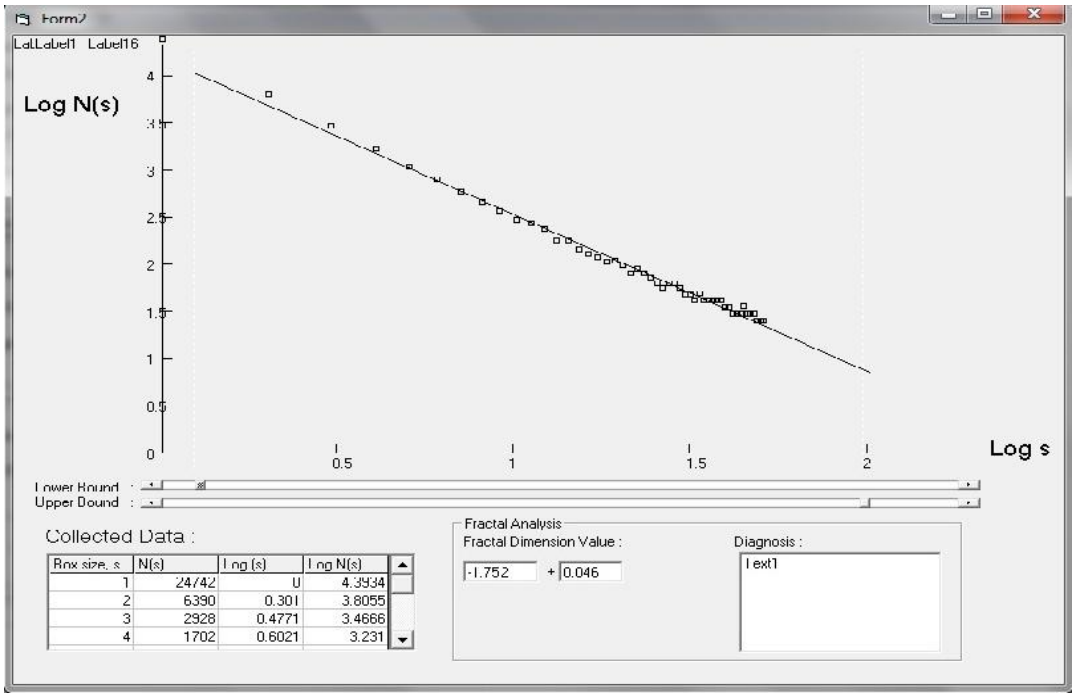


Figure 4.7(i): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.752 for the image of Figure 4.2(i)

Table 4.21: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.2 (i)

Box size,s	N(s)	Log (s)	Log N(s)
1	24742	0	4.3934
2	6390	0.301	3.8055
3	2928	0.4771	3.4666
4	1702	0.6021	3.2310
5	1475	0.699	3.1688
8	1073	0.7782	3.0306
7	818	0.8451	2.9128
8	647	0.8031	2.8112
8	527	0.9542	2.7216
10	438	1	2.6414
11	371	1.0414	2.5689
12	318	1.0782	2.5027
13	277	1.1138	2.4418
14	243	1.1481	2.3854
15	215	1.1781	2.3329
16	192	1.2041	2.2838
17	173	1.2304	2.2377
18	156	1.2553	2.1942
19	142	1.2788	2.1530
20	130	1.301	2.1140
21	119	1.3222	2.0769
22	110	1.3424	2.0415
23	102	1.3617	2.0077
24	94	1.3802	1.9753
25	88	1.3878	1.9442

Box size,s	N(s)	Log (s)	Log N(s)
26	82	1.415	1.9144
27	77	1.4314	1.8857
28	72	1.4472	1.8580
29	68	1.4624	1.8313
30	64	1.4771	1.8055
31	60	1.4814	1.7805
32	57	1.5051	1.7564
33	54	1.5185	1.7330
34	51	1.5315	1.7102
35	49	1.5441	1.6882
36	46	1.5563	1.6668
37	44	1.5682	1.6459
38	42	1.5798	1.6256
39	40	1.5811	1.6059
40	39	1.6021	1.5866
41	37	1.6128	1.5678
42	35	1.6232	1.5495
43	34	1.6335	1.5316
44	33	1.6435	1.5141
45	31	1.6532	1.4970
46	30	1.6628	1.4802
47	29	1.6721	1.4639
48	28	1.6812	1.4479
49	27	1.6902	1.4322
50	26	1.699	1.4168

The data for box count dimension of the images of Figure 4.3(a) to (i) are tabulated in Table 4.22 to 4.30 with their screen shots of the graphs which yield their fractal dimension values in Figure 4.8(a) to (i) respectively.

Table 4.22: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (a)

Box size,s	N(s)	Log (s)	Log N(s)
1	27900	0	4.4456
2	7144	0.301	3.9408
3	3276	0.4771	3.6455
4	1872	0.6021	3.4359
5	1877	0.699	3.2734
6	1382	0.7782	3.1406
7	1068	0.8451	3.0284
8	853	0.8031	2.9311
9	700	0.9542	2.8453
10	587	1	2.7686
11	500	1.0414	2.6992
12	432	1.0782	2.6358
13	378	1.1138	2.5775
14	334	1.1481	2.5235
15	297	1.1781	2.4733
16	267	1.2041	2.4263
17	241	1.2304	2.3821
18	219	1.2553	2.3405
19	200	1.2788	2.3011
20	184	1.301	2.2638
21	169	1.3222	2.2282
22	156	1.3424	2.1944
23	145	1.3617	2.1620
24	135	1.3802	2.1310
25	126	1.3878	2.1013

Box size,s	N(s)	Log (s)	Log N(s)
26	118	1.415	2.0727
27	111	1.4314	2.0452
28	104	1.4472	2.0187
29	98	1.4624	1.9932
30	93	1.4771	1.9685
31	88	1.4814	1.9446
32	83	1.5051	1.9215
33	79	1.5185	1.8991
34	75	1.5315	1.8773
35	72	1.5441	1.8562
36	68	1.5563	1.8357
37	65	1.5682	1.8157
38	63	1.5798	1.7963
39	60	1.5811	1.7774
40	57	1.6021	1.7589
41	55	1.6128	1.7410
42	53	1.6232	1.7234
43	51	1.6335	1.7063
44	49	1.6435	1.6895
45	47	1.6532	1.6732
46	45	1.6628	1.6572
47	44	1.6721	1.6415
48	42	1.6812	1.6262
49	41	1.6902	1.6111
50	39	1.699	1.5964

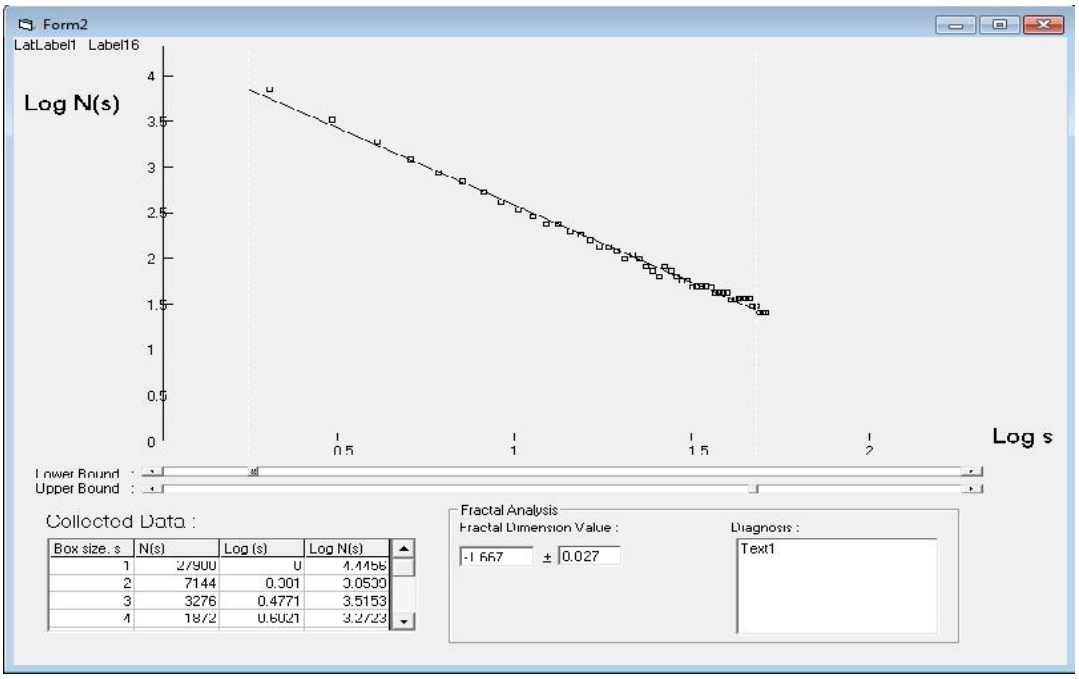


Figure 4.8(a): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.667 for the image of Figure 4.3(a)

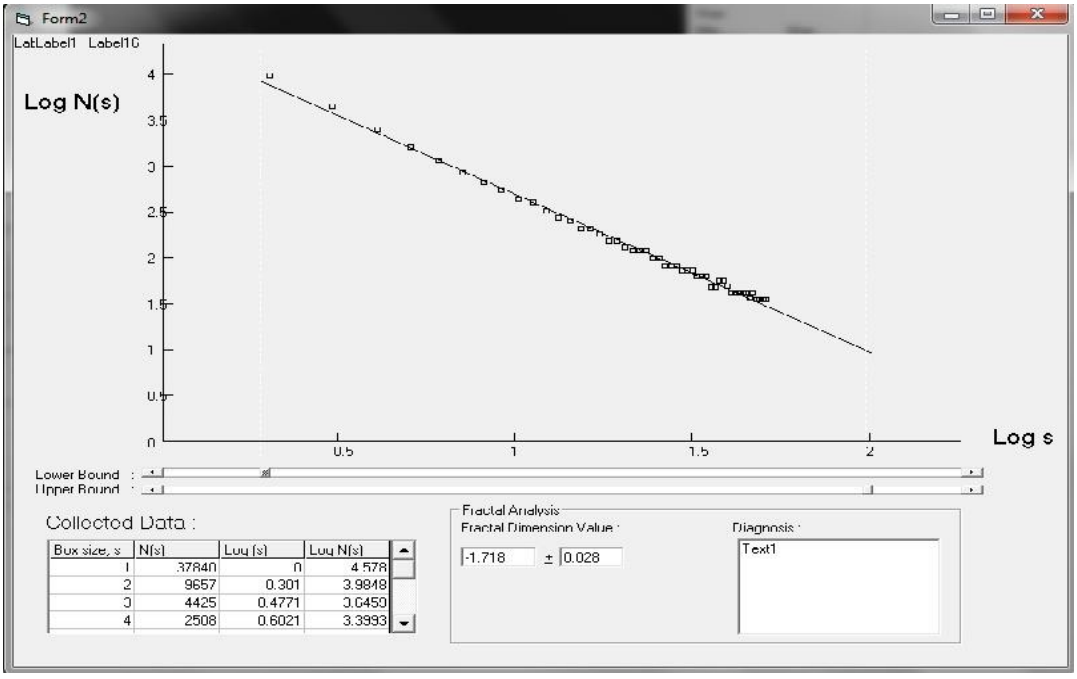


Figure 4.8(b): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.718 for the image of Figure 4.3(b)

Table 4.23: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (b)

Box size,s	N(s)	Log (s)	Log N(s)
1	37840	0	4.5780
2	9657	0.301	3.9848
3	4425	0.4771	3.6459
4	2508	0.6021	3.3993
5	2383	0.699	3.3772
8	1742	0.7782	3.2411
7	1620	0.8451	3.2095
8	1140	0.8031	3.0569
8	858	0.9542	2.9335
10	667	1	2.8241
11	546	1.0414	2.7372
12	703	1.0782	2.8470
13	612	1.1138	2.7864
14	537	1.1481	2.7303
15	477	1.1781	2.6781
16	426	1.2041	2.6292
17	383	1.2304	2.5833
18	347	1.2553	2.5401
19	316	1.2788	2.4991
20	289	1.301	2.4603
21	265	1.3222	2.4234
22	244	1.3424	2.3882
23	226	1.3617	2.3545
24	210	1.3802	2.3223
25	196	1.3878	2.2914

Box size,s	N(s)	Log (s)	Log N(s)
26	140	1.415	2.1471
27	131	1.4314	2.1189
28	124	1.4472	2.0918
29	116	1.4624	2.0656
30	110	1.4771	2.0403
31	104	1.4814	2.0158
32	98	1.5051	1.9922
33	93	1.5185	1.9692
34	88	1.5315	1.9469
35	84	1.5441	1.9253
36	80	1.5563	1.9043
37	77	1.5682	1.8838
38	73	1.5798	1.8639
39	70	1.5811	1.8446
40	67	1.6021	1.8257
41	64	1.6128	1.8072
42	62	1.6232	1.7893
43	59	1.6335	1.7717
44	57	1.6435	1.7545
45	55	1.6532	1.7378
46	53	1.6628	1.7214
47	51	1.6721	1.7053
48	49	1.6812	1.6896
49	47	1.6902	1.6742
50	46	1.699	1.6592

Table 4.24: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (c)

Box size,s	N(s)	Log (s)	Log N(s)
1	4050	0	3.6075
2	1064	0.301	3.0269
3	540	0.4771	2.7324
4	300	0.6021	2.4771
5	260	0.699	2.4151
8	191	0.7782	2.2800
7	146	0.8451	2.1658
8	117	0.8031	2.0668
8	95	0.9542	1.9796
10	80	1	1.9015
11	68	1.0414	1.8309
12	58	1.0782	1.7664
13	51	1.1138	1.7071
14	45	1.1481	1.6522
15	40	1.1781	1.6011
16	36	1.2041	1.5533
17	32	1.2304	1.5084
18	29	1.2553	1.4660
19	27	1.2788	1.4259
20	24	1.301	1.3879
21	22	1.3222	1.3518
22	21	1.3424	1.3173
23	19	1.3617	1.2844
24	18	1.3802	1.2529
25	17	1.3878	1.2226

Box size,s	N(s)	Log (s)	Log N(s)
26	16	1.415	1.1936
27	15	1.4314	1.1656
28	14	1.4472	1.1386
29	13	1.4624	1.1126
30	12	1.4771	1.0875
31	12	1.4814	1.0632
32	11	1.5051	1.0397
33	10	1.5185	1.0169
34	10	1.5315	0.9948
35	9	1.5441	0.9733
36	9	1.5563	0.9524
37	9	1.5682	0.9321
38	8	1.5798	0.9124
39	8	1.5811	0.8931
40	7	1.6021	0.8744
41	7	1.6128	0.8561
42	7	1.6232	0.8382
43	7	1.6335	0.8208
44	6	1.6435	0.8038
45	6	1.6532	0.7871
46	6	1.6628	0.7708
47	6	1.6721	0.7549
48	5	1.6812	0.7393
49	5	1.6902	0.7240
50	5	1.699	0.7091

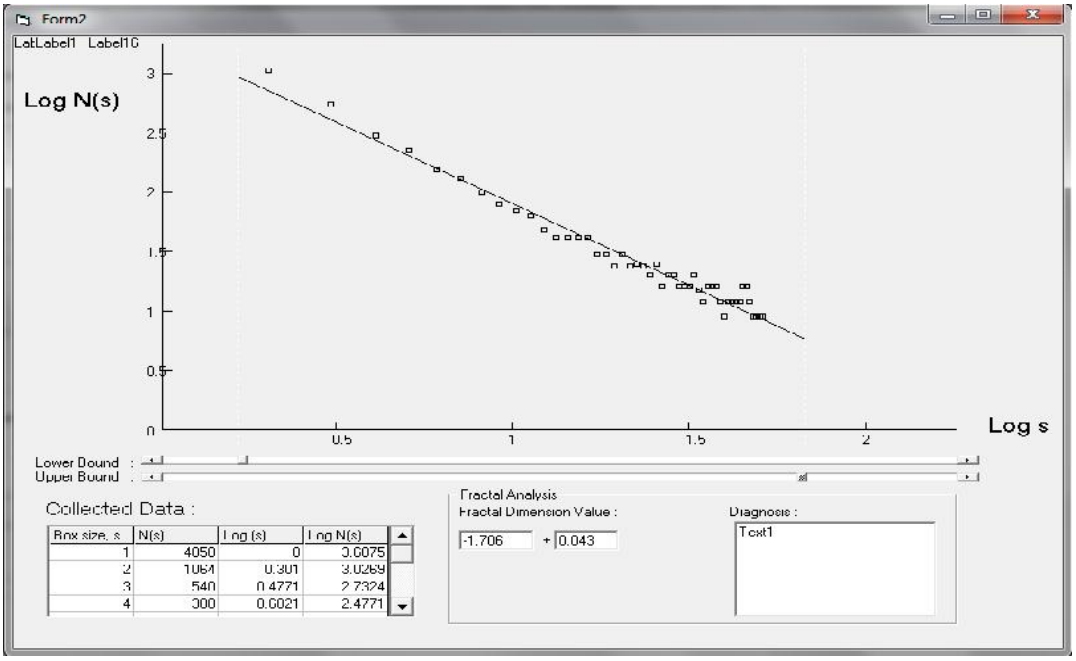


Figure 4.8(c): Screen shot image of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.706 for the image of Figure 4.3(c)

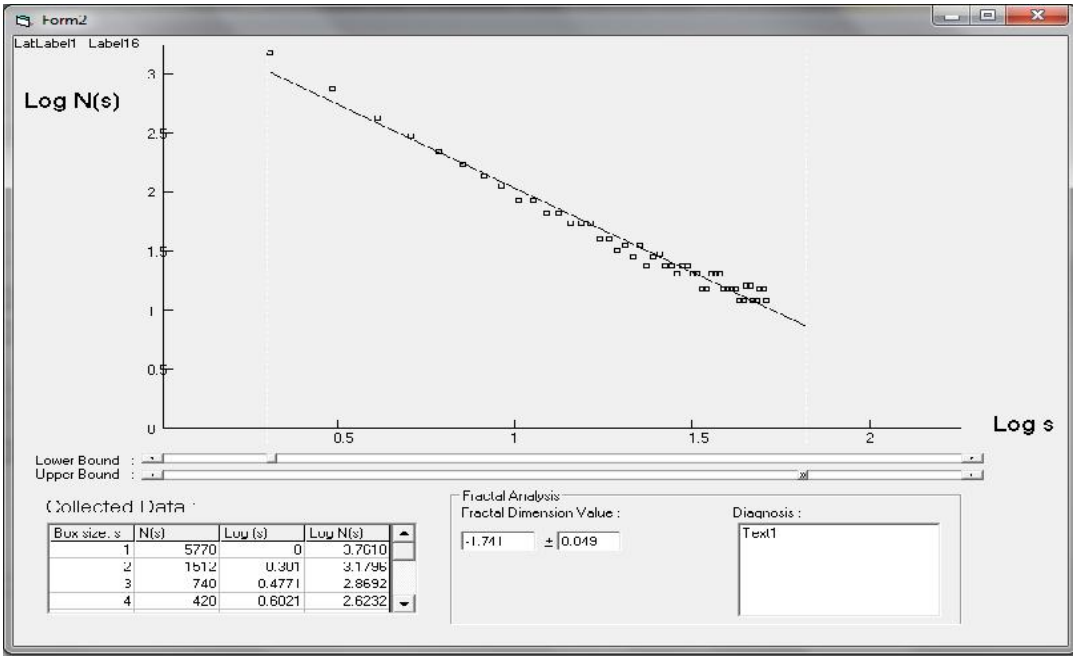


Figure 4.8(d): Screen shot image of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.741 for the image of Figure 4.3(d)

Table 4.25: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (d)

Box size,s	N(s)	Log (s)	Log N(s)
1	5778	0	3.7618
2	1512	0.301	3.1796
3	740	0.4771	2.8692
4	420	0.6021	2.6232
5	351	0.699	2.5449
8	255	0.7782	2.4070
7	195	0.8451	2.2905
8	155	0.8031	2.1895
8	126	0.9542	2.1005
10	105	1	2.0208
11	89	1.0414	1.9487
12	76	1.0782	1.8829
13	66	1.1138	1.8224
14	58	1.1481	1.7664
15	52	1.1781	1.7142
16	46	1.2041	1.6654
17	42	1.2304	1.6196
18	38	1.2553	1.5764
19	34	1.2788	1.5355
20	31	1.301	1.4967
21	29	1.3222	1.4598
22	27	1.3424	1.4246
23	25	1.3617	1.3910
24	23	1.3802	1.3589
25	21	1.3878	1.3280

Box size,s	N(s)	Log (s)	Log N(s)
26	20	1.415	1.2983
27	19	1.4314	1.2698
28	17	1.4472	1.2423
29	16	1.4624	1.2158
30	15	1.4771	1.1901
31	15	1.4814	1.1653
32	14	1.5051	1.1413
33	13	1.5185	1.1181
34	12	1.5315	1.0955
35	12	1.5441	1.0736
36	11	1.5563	1.0523
37	11	1.5682	1.0316
38	10	1.5798	1.0114
39	10	1.5811	0.9918
40	9	1.6021	0.9726
41	9	1.6128	0.9539
42	9	1.6232	0.9357
43	8	1.6335	0.9179
44	8	1.6435	0.9005
45	8	1.6532	0.8836
46	7	1.6628	0.8669
47	7	1.6721	0.8507
48	7	1.6812	0.8348
49	7	1.6902	0.8192
50	6	1.699	0.8039

Table 4.26: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (e)

Box size,s	N(s)	Log (s)	Log N(s)
1	16608	0	4.2203
2	4263	0.301	3.6297
3	1947	0.4771	3.2894
4	1170	0.6021	3.0682
5	1054	0.699	3.0230
8	771	0.7782	2.8873
7	592	0.8451	2.7726
8	471	0.8031	2.6733
8	385	0.9542	2.5857
10	322	1	2.5073
11	273	1.0414	2.4364
12	235	1.0782	2.3717
13	205	1.1138	2.3121
14	181	1.1481	2.2570
15	161	1.1781	2.2057
16	144	1.2041	2.1576
17	130	1.2304	2.1125
18	117	1.2553	2.0700
19	107	1.2788	2.0298
20	98	1.301	1.9916
21	90	1.3222	1.9553
22	83	1.3424	1.9207
23	77	1.3617	1.8877
24	72	1.3802	1.8560
25	67	1.3878	1.8256

Box size,s	N(s)	Log (s)	Log N(s)
26	63	1.415	1.7965
27	59	1.4314	1.7684
28	55	1.4472	1.7413
29	52	1.4624	1.7152
30	49	1.4771	1.6900
31	46	1.4814	1.6656
32	44	1.5051	1.6420
33	42	1.5185	1.6191
34	40	1.5315	1.5969
35	38	1.5441	1.5753
36	36	1.5563	1.5544
37	34	1.5682	1.5340
38	33	1.5798	1.5141
39	31	1.5811	1.4948
40	30	1.6021	1.4760
41	29	1.6128	1.4576
42	28	1.6232	1.4397
43	26	1.6335	1.4222
44	25	1.6435	1.4051
45	24	1.6532	1.3883
46	24	1.6628	1.3720
47	23	1.6721	1.3560
48	22	1.6812	1.3403
49	21	1.6902	1.3250
50	20	1.699	1.3100

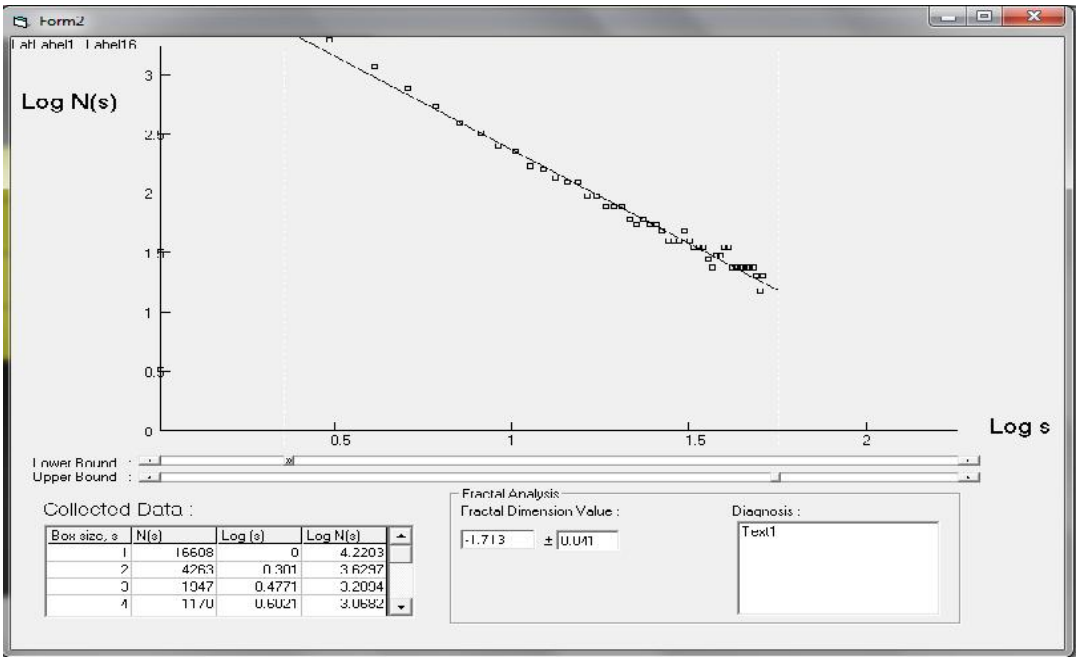


Figure 4.8(e): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.713 for the image of Figure 4.3(e)

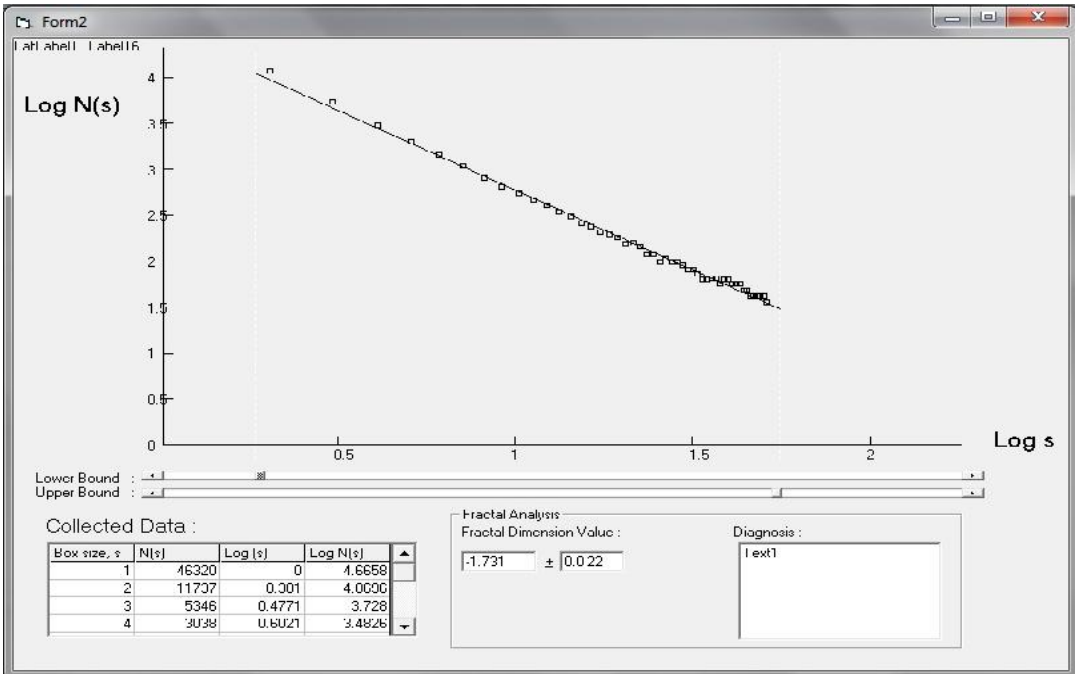


Figure 4.8(f): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.731 for the image of Figure 4.3(f)

Table 4.27: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (f)

Box size,s	N(s)	Log (s)	Log N(s)
1	46320	0	4.6658
2	11737	0.301	4.0608
3	5346	0.4771	3.7583
4	3038	0.6021	3.5437
5	2009	0.699	3.3772
8	1742	0.7782	3.2411
7	1620	0.8451	3.1261
8	1140	0.8031	3.0265
8	858	0.9542	2.9386
10	667	1	2.8600
11	546	1.0414	2.7889
12	703	1.0782	2.7240
13	612	1.1138	2.6642
14	537	1.1481	2.6090
15	477	1.1781	2.5575
16	426	1.2041	2.5093
17	383	1.2304	2.4641
18	347	1.2553	2.4214
19	316	1.2788	2.3811
20	289	1.301	2.3428
21	265	1.3222	2.3064
22	244	1.3424	2.2717
23	226	1.3617	2.2386
24	210	1.3802	2.2068
25	196	1.3878	2.1763

Box size,s	N(s)	Log (s)	Log N(s)
26	165	1.415	2.2165
27	154	1.4314	2.1881
28	145	1.4472	2.1608
29	136	1.4624	2.1344
30	128	1.4771	2.1089
31	121	1.4814	2.0843
32	115	1.5051	2.0604
33	109	1.5185	2.0373
34	103	1.5315	2.0148
35	98	1.5441	1.9930
36	94	1.5563	1.9718
37	89	1.5682	1.9512
38	85	1.5798	1.9312
39	82	1.5811	1.9117
40	78	1.6021	1.8926
41	75	1.6128	1.8741
42	72	1.6232	1.8560
43	69	1.6335	1.8383
44	66	1.6435	1.8210
45	64	1.6532	1.8041
46	61	1.6628	1.7876
47	59	1.6721	1.7714
48	57	1.6812	1.7556
49	55	1.6902	1.7401
50	53	1.699	1.7249

Table 4.28: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (g)

Box size,s	N(s)	Log (s)	Log N(s)
1	6290	0	3.7987
2	1634	0.301	3.2133
3	780	0.4771	2.8921
4	460	0.6021	2.6628
5	323	0.699	2.5092
8	224	0.7782	2.3502
7	221	0.8451	2.3450
8	176	0.8031	2.2456
8	144	0.9542	2.1579
10	120	1	2.0795
11	102	1.0414	2.0086
12	88	1.0782	1.9438
13	77	1.1138	1.8842
14	67	1.1481	1.8290
15	60	1.1781	1.7777
16	54	1.2041	1.7296
17	48	1.2304	1.6845
18	44	1.2553	1.6420
19	40	1.2788	1.6017
20	37	1.301	1.5635
21	34	1.3222	1.5272
22	31	1.3424	1.4926
23	29	1.3617	1.4595
24	27	1.3802	1.4278
25	25	1.3878	1.3974

Box size,s	N(s)	Log (s)	Log N(s)
26	21	1.415	1.3239
27	20	1.4314	1.2952
28	19	1.4472	1.2676
29	17	1.4624	1.2410
30	16	1.4771	1.2152
31	15	1.4814	1.1903
32	15	1.5051	1.1662
33	14	1.5185	1.1428
34	13	1.5315	1.1201
35	13	1.5441	1.0981
36	12	1.5563	1.0767
37	11	1.5682	1.0559
38	11	1.5798	1.0357
39	10	1.5811	1.0159
40	10	1.6021	0.9967
41	10	1.6128	0.9779
42	9	1.6232	0.9596
43	9	1.6335	0.9418
44	8	1.6435	0.9243
45	8	1.6532	0.9072
46	8	1.6628	0.8905
47	7	1.6721	0.8742
48	7	1.6812	0.8582
49	7	1.6902	0.8425
50	7	1.699	0.8272

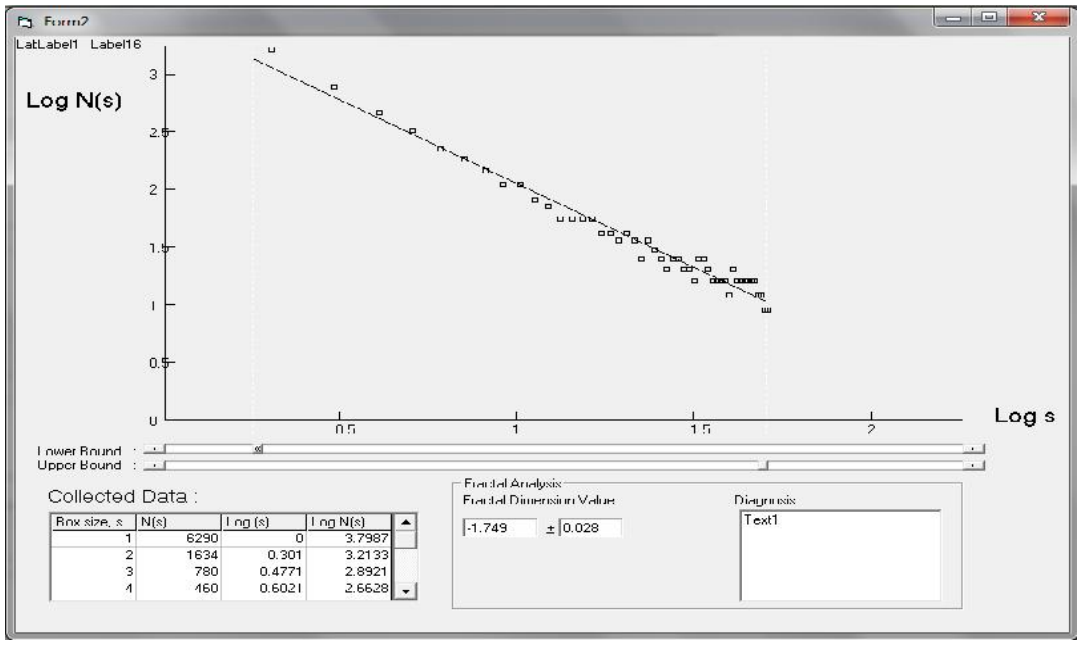


Figure 4.8(g): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.749 for the image of Figure 4.3(g)

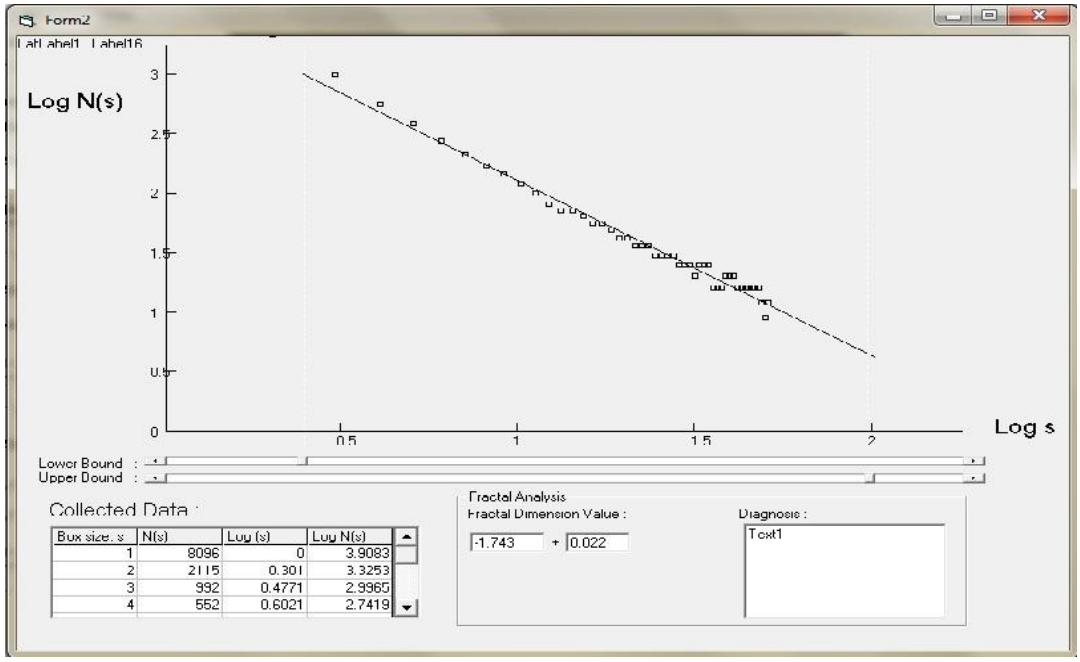


Figure 4.8(h): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.743 for the image of Figure 4.3(h)

Table 4.29: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (h)

Box size,s	N(s)	Log (s)	Log N(s)
1	8096	0	3.9083
2	2115	0.301	3.3253
3	992	0.4771	2.9965
4	552	0.6021	2.7419
5	490	0.699	2.6900
8	356	0.7782	2.5520
7	272	0.8451	2.4353
8	216	0.8031	2.3342
8	176	0.9542	2.2451
10	146	1	2.1653
11	124	1.0414	2.0932
12	106	1.0782	2.0273
13	93	1.1138	1.9667
14	81	1.1481	1.9106
15	72	1.1781	1.8584
16	64	1.2041	1.8095
17	58	1.2304	1.7636
18	53	1.2553	1.7204
19	48	1.2788	1.6794
20	44	1.301	1.6406
21	40	1.3222	1.6037
22	37	1.3424	1.5685
23	34	1.3617	1.5348
24	32	1.3802	1.5026
25	30	1.3878	1.4717
26	28	1.415	1.4420
27	26	1.4314	1.4134
28	24	1.4472	1.3859
29	23	1.4624	1.3593
30	22	1.4771	1.3337
31	20	1.4814	1.3089
32	19	1.5051	1.2848
33	18	1.5185	1.2615
34	17	1.5315	1.2389
35	16	1.5441	1.2170
36	16	1.5563	1.1957
37	15	1.5682	1.1749
38	14	1.5798	1.1547
39	14	1.5811	1.1351
40	13	1.6021	1.1159
41	13	1.6128	1.0972
42	12	1.6232	1.0790
43	12	1.6335	1.0612
44	11	1.6435	1.0438
45	11	1.6532	1.0268
46	10	1.6628	1.0101
47	10	1.6721	0.9938
48	10	1.6812	0.9779
49	9	1.6902	0.9623
50	9	1.699	0.9470

Table 4.30: The number of box-count, N(s) with respect of grid length of square meshes, s for the image of Figure 4.3 (i)

Box size,s	N(s)	Log (s)	Log N(s)
1	8820	0	3.9455
2	2322	0.301	3.4166
3	1073	0.4771	3.1072
4	644	0.6021	2.8877
5	522	0.699	2.7174
8	379	0.7782	2.5783
7	289	0.8451	2.4607
8	228	0.8031	2.3588
8	186	0.9542	2.2689
10	154	1	2.1885
11	131	1.0414	2.1158
12	112	1.0782	2.0494
13	97	1.1138	1.9883
14	85	1.1481	1.9318
15	76	1.1781	1.8791
16	68	1.2041	1.8299
17	61	1.2304	1.7836
18	55	1.2553	1.7400
19	50	1.2788	1.6987
20	46	1.301	1.6596
21	42	1.3222	1.6224
22	39	1.3424	1.5869
23	36	1.3617	1.5529
24	33	1.3802	1.5205
25	31	1.3878	1.4893
26	29	1.415	1.4594
27	27	1.4314	1.4306
28	25	1.4472	1.4028
29	24	1.4624	1.3761
30	22	1.4771	1.3502
31	21	1.4814	1.3252
32	20	1.5051	1.3010
33	19	1.5185	1.2775
34	18	1.5315	1.2547
35	17	1.5441	1.2326
36	16	1.5563	1.2111
37	15	1.5682	1.1902
38	15	1.5798	1.1698
39	14	1.5811	1.1500
40	14	1.6021	1.1307
41	13	1.6128	1.1118
42	12	1.6232	1.0935
43	12	1.6335	1.0755
44	11	1.6435	1.0580
45	11	1.6532	1.0408
46	11	1.6628	1.0240
47	10	1.6721	1.0076
48	10	1.6812	0.9916
49	9	1.6902	0.9758
50	9	1.699	0.9604

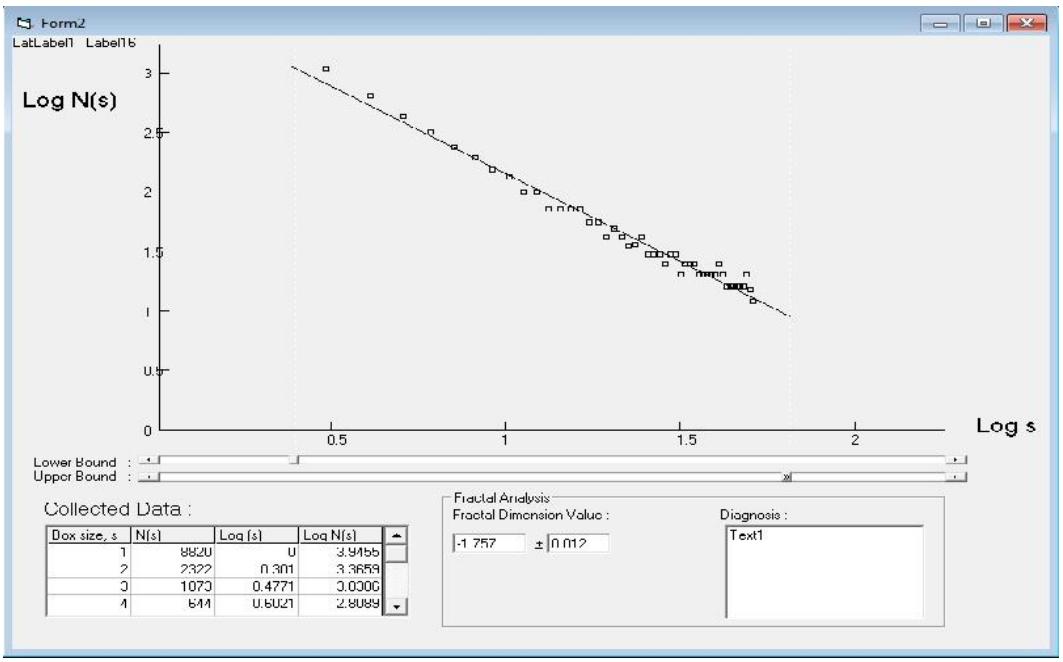


Figure 4.8(i): Screen shot of the graph of log *N*(*s*) vs. log *s* (linear scale) with the calculated fractal (box) dimension, D = 1.757 for the image of Figure 4.3(i)

The fractal dimension values of all of the experimentally obtained fractals are listed in Table 4.31- 4.33. The tables show that the fractal dimension values are ~1.7 which is typical of diffusion-limited aggregates (Chandra, 1996). Furthermore, the fractal patterns are similar to the computer simulated diffusion-limited aggregate patterns reported by Ossadnik et al. (1994).

Table 4.31: Fractal dimension values for the experimentally cultured fractals in Chitosan-$AgNO_3$ film as shown in Figure 4.1

Experimentally cultured fractals	
Figure	Fractal dimension
 4.1(a)	1.705 ± 0.035
 4.1(b)	1.714 ± 0.044
 4.1(c)	1.726 ± 0.049
 4.1(d)	1.707 ± 0.053
 4.1(e)	1.698 ± 0.036
 4.1(f)	1.709 ± 0.045
 4.1(g)	1.731 ± 0.051

Table 4.31 continued...

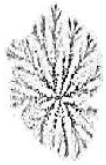
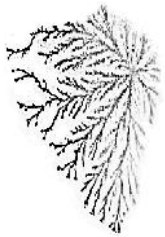
 4.1(h)	1.729 ± 0.043
 4.1(i)	1.774 ± 0.037
 4.1(j)	1.761 ± 0.015

Table 4.32: Fractal dimension values for the experimentally cultured fractals in PEO-NH_4I film as shown in Figure 4.2

Experimentally cultured fractals	
Figure	Fractal dimension
 4.2(a)	1.709 ± 0.053
 4.2(b)	1.728 ± 0.031

Table 4.32 continued...

 4.2(c)	1.714 ± 0.015
 4.2(d)	1.753 ± 0.045
 4.2(e)	1.768 ± 0.047
 4.2(f)	1.741 ± 0.039
 4.2(g)	1.714 ± 0.043
 4.2(h)	1.746 ± 0.034
 4.2(i)	1.752 ± 0.046

Table 4.32 continued...

 4.2(j)	1.794± 0.043

Table 4.33: Fractal dimension values for the experimentally cultured fractals in PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film as shown in Figure 4.3

Experimentally cultured fractals	
Figure	Fractal dimension
 4.3(a)	1.667 ± 0.027
 4.3(b)	1.718 ± 0.028
 4.3(c)	1.706 ± 0.043
 4.3(d)	1.741 ± 0.049
 4.3(e)	1.713 ± 0.041

Table 4.33 continued...

 4.3(f)	1.731 ± 0.022
 4.3(g)	1.749 ± 0.028
 4.3(h)	1.743 ± 0.023
 4.3(i)	1.757 ± 0.012
 4.3(j)	1.786 ± 0.044

Antecedently, as explained in Section 4.1.1, the fractals formed at different nucleation centers and then grew in the direction away from the nucleation sites. The fractals do not overlap each other but are separated from each other by a definite boundary. Neighboring fractals seem to have an effect to each other's definite pattern of advance. Thus it is important to also consider that the fractal patterns observed in every polymer electrolyte film can be viewed as multiple clusters formed within particular areas. Figure 4.9 shows three identified regions of multicluster fractals as observed in the chitosan-$AgNO_3$ film with each region consisting of 3 to 6 fractal clusters. There are 6 fractal clusters in region

(a) with the biggest cluster placed in the center surrounded by smaller fractal clusters and 3 fractal clusters of different sizes in regions (b) and (c) each. As with the earlier single cluster fractal patterns discussed in the beginning of this Section, the fractal dimension of every individual fractal of the multicluster fractals are exhibited in Table 4.34.

Figure 4.9 : Different areas of multicluster fractal patterns in regions (a), (b) and (c) as observed in the chitosan-$AgNO_3$ film

Table 4.34: The fractal dimension of every individual fractal of the multicluster fractal patterns in regions (a), (b) and (c) as observed in the chitosan-$AgNO_3$ film

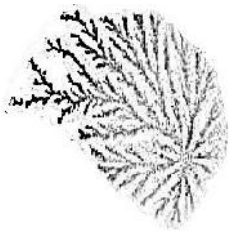
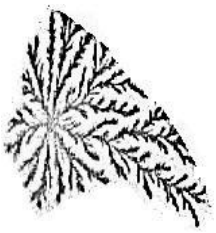
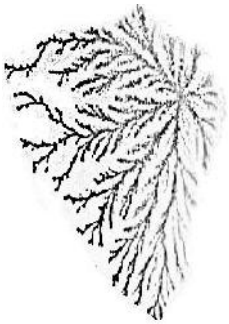
Region	A	B	c
Fractal dimension values	 a1) 1.707 ± 0.042	 b1) 1.726 ± 0.049	 c1) 1.774 ± 0.037
	 a2) 1.718 ± 0.047-		

Table 4.34 continued...

	a3) 1.722 ± 0.038	 b2) 1.789 ± 0.036	 c2) 1.698 ± 0.036
	a4) 1.725 ± 0.043		
	a5) 1.736 ± 0.041	 b3) 1.791 ± 0.033	 c3) 1.761 ± 0.015
	 a6) 1.779 ± 0.045		

Figures 4.10 and 4.11 show identified regions of multicluster fractals observed in the PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 films respectively. In the PEO-NH_4I film, there are 3 fractal clusters of different sizes in each region. While in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 film (Figure 4.11) only two clearly apparent multicluster fractals of 4 and 6 individual clusters in each region (a) and (b) respectively. The fractal dimensions of every individual fractal of the multicluster fractals ascertained in the PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 films are listed in Table 4.35 and Table 4.36 respectively.

Figure 4.10 : Different areas of multicluster fractal patterns in regions (a), (b) and (c) as observed in the PEO-NH_4I film

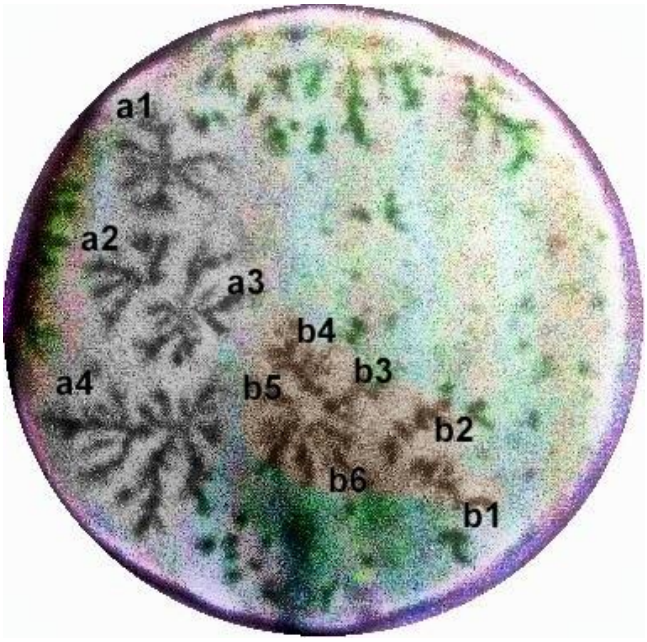


Figure 4.11 : Different areas of multicluster fractal patterns in regions (a) and (b) as observed in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 film

Table 4.35: The fractal dimension of every individual fractal of the multicluster fractal patterns in regions (a), (b) and (c) as observed in the PEO-NH_4I film

Region	a	b	c
Fractal dimension values	 a1) 1.709 ± 0.053	 b1) 1.741 ± 0.039	 c1) 1.707 ± 0.044
	 a2) 1.714 ± 0.043	 b2) 1.752 ± 0.046	 c2) 1.726 ± 0.042
	 a3) 1.746 ± 0.034	 b3) 1.794 ± 0.043	 c3) 1.768 ± 0.047

Generally, from all the images of the cultured fractals of the identified regions of multicluster fractal patterns, as the cluster grows larger the fractal dimension value becomes higher. Comparison of the fractal dimension values of the cultured fractals and their respected simulated ones is presented in Chapter 6.

Table 4.36: The fractal dimension of every individual fractal of the multicluster fractal patterns in regions (a) and (b) as observed in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$ dispersed with Cr_2O_3 film

Region	a	b
Fractal dimension values	 a1) 1.749 ± 0.028	 b1) 1.718 ± 0.045
	 a2) 1.743 ± 0.023	 b2) 1.707 ± 0.042
	 a3) 1.757 ± 0.012	 b3) 1.706 ± 0.043
		 b4) 1.718 ± 0.028
	 a4) 1.786 ± 0.044	 b5) 1.741 ± 0.049
		 b6) 1.720 ± 0.043

C H A P T E R 5

RESULTS AND DISCUSSIONS OF SIMULATED SINGLE CLUSTER FRACTALS

In this chapter, results of the simulated fractals are presented and discussed. Simulation works on single cluster type of fractal growth patterns were carried out by applying certain rules for each type of simulation. Further discussions on the rules and conditions applied in every simulation can be found in the following sections. Consequently, comparisons of the simulated fractal patterns were made with original fractals observed in polymer films.

5.1 Simulation Program of Single Cluster Fractal Growth Pattern

For simulation of single cluster fractal growth pattern, the traditional DLA model was applied. DLA is modeled by tracking random walking particles and a cluster to which the random walking particles may stick. The cluster is composed of a seed, the initial particle member, and other particles that have stuck to the cluster. The random walkers are released one by one, only one on the lattice at a time, to wander around until they land in a position touching a cluster member as shown in Figure 5.1. If it touches, it sticks. Then the next particle is released. The details of particle release and sticking are dependent on the practical limitations of the simulation.

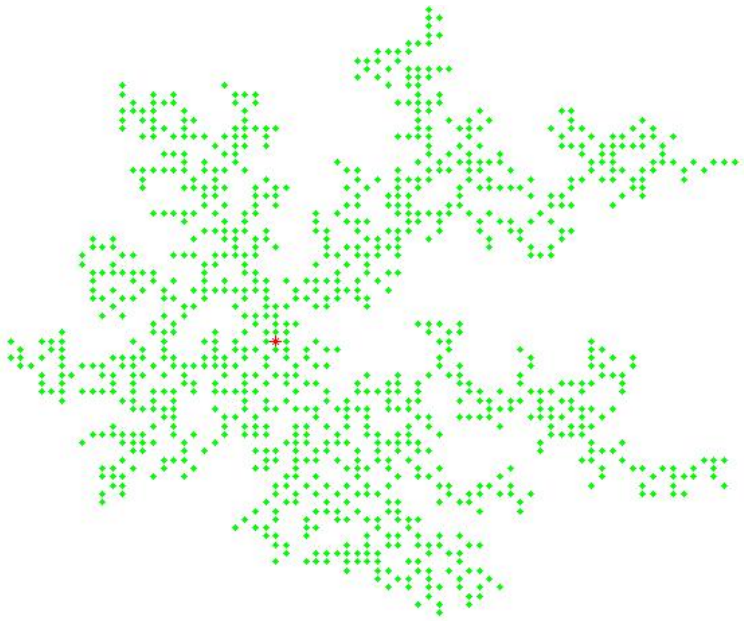


Figure 5.1: A typical simulation of single cluster DLA type fractals

Specifically, the simulation program developed was uniquely designed to enable a user to observe every stage of growth of any simulated single cluster of fractal growth pattern. This was done by implementing a pause button that will pause running of the simulation at any time of execution.

In this work, the simulation of the fractal patterns on variable numbers of lattice sites of 4, 6 and 8 has been carried out and the results obtained from the simulations were analyzed. In every simulation, a fractal cluster of required size represented by max particles, M was simulated within a specific radius, R and their fractal dimension was calculated simultaneously. The simulation for a single cluster of fractals was also done by setting the desired value of the sticking coefficient, α within the range of 0 to 1. For large enough size fractals, the number of max particles chosen here was M = 5000. The selection of higher number of sticking coefficient generally resulted in less dense branches of fractal cluster.

From the analysis run on the simulation model obtained, there are some properties of the clusters that have been identified. These properties are as follows:

1. Branching and screening

The random growth process leads to the formation of small tips which are likely to capture diffusing particles. They screen their surroundings which later have the effect of screening that self-stabilizes the tip until it grows even larger forming new tips, branched and tree–like object.

2. Scale invariance, lack of a typical length-scale

As the stage of growth increased, it seems like they were hierarchy of arms, branches, twigs and sprouts with fjord like empty regions of all sizes.

3. Stochastic self–similarity

Substructures of the clusters look pretty much alike, i.e. statistical properties are reproduced after proper rescaling.

Apart from these properties of the DLA cluster, it was also discovered that diffusing particles are very unlikely to wander into one of the inner fjords. The diffusing particles had a high probability to attach to the protruding tips. They were easily detected from the formation of most recent particles and growth occurred essentially only in a small active zone within the predetermined radius. Generally, for initial simulation work, the simulation parameters for all three fractals in chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films were specifically chosen based on the common properties of every fractals observed in the films. The simulated single cluster fractal aggregate of the chitosan-$AgNO_3$ film is depicted in Figure 5.2. The figure also illustrates

the log *M* vs. log *R* (*M*) plots of the simulated fractals which is vital in the calculation of fractal dimension.

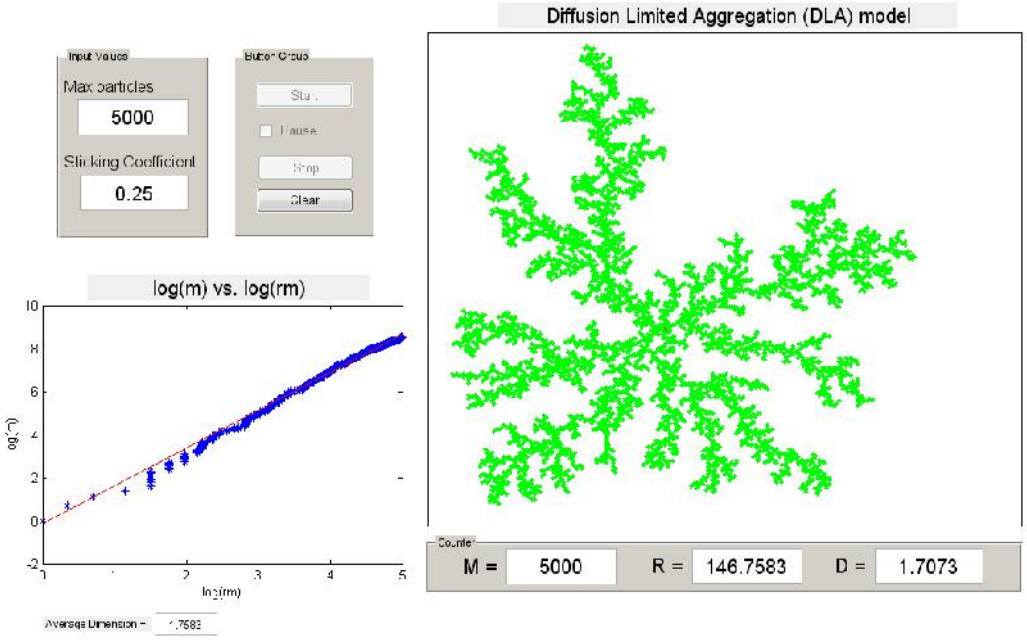


Figure 5.2: Simulation of single cluster fractal found in Chitosan-$AgNo_3$ films

As shown in Figure 5.2, the simulation of single cluster fractal found in Chitosan-$AgNO_3$ films was done by choosing the input values; Max particles:5000 and Sticking coefficient:0.25 with the final radius, *R* achieved is 146.76 with fractal dimension value of 1.7073.

In spite of the usefulness of the fractal analysis done with the above simulation, it is actually better to consider in real condition that a single cluster fractal has certain properties such as shape of a single cluster and denser branches. Thus it is essential to further consider the sticking probability of any of the near neighboring particles by introducing a sticking coefficient. The effects of introducing certain values of sticking coefficient can influence the nature of the density of any single cluster fractal pattern

withal the importance of taking into account the number of lattice sites which have the effect on the general shape of a cluster.

Figures 5.3-5.5 depict simulations of a single cluster fractal aggregate of the chitosan-$AgNO_3$ film with combinations of values of number of particles, sticking coefficient and number of lattice sites.

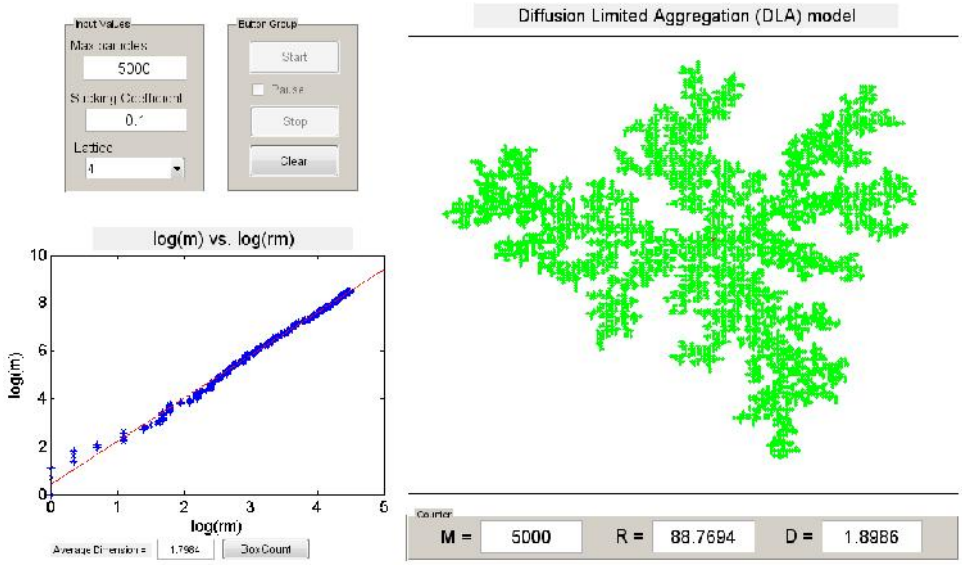


Figure 5.3(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.1 and 4 lattice site

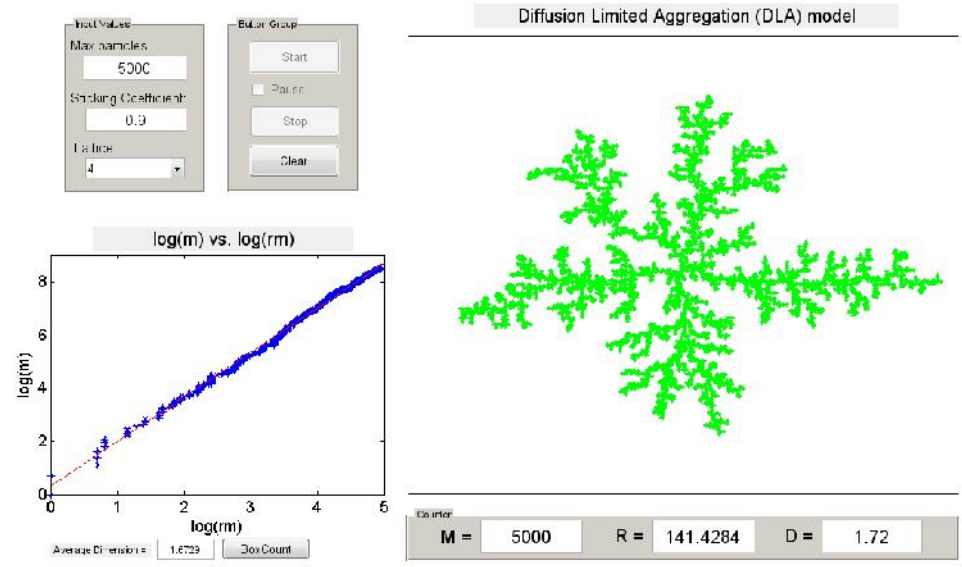


Figure 5.3(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.9 and 4 lattice site

In Figure 5.3(a) and (b), the simulation of single cluster fractal found in Chitosan-$AgNO_3$ films was done by choosing the input values; Max particles:5000, 4 lattice sites and Sticking coefficient:0.1 and 0.9 with the final radius, *R* achieved are 88.77 and 141.43 with fractal dimension value of 1.8986 and 1.72 respectively. In both figures, the former has much denser branches compared to the latter caused by the lower value of sticking coefficient.

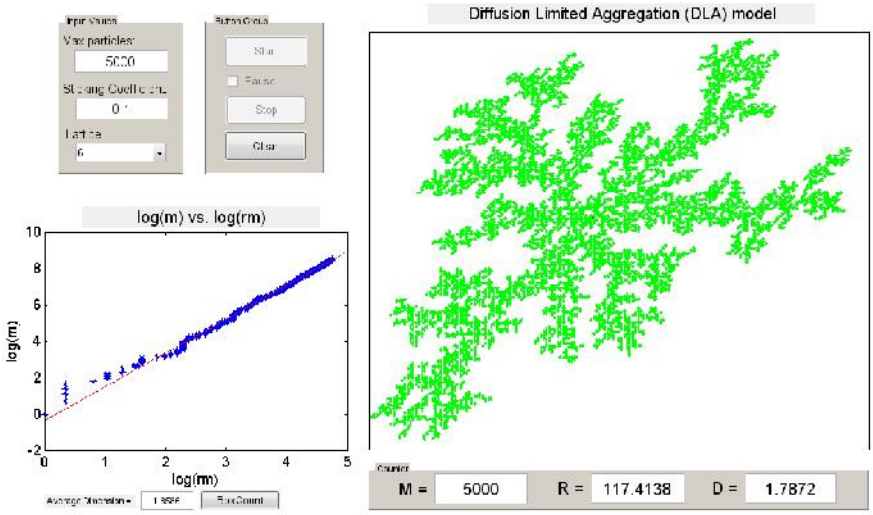


Figure 5.4(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.1 and 6 lattice site

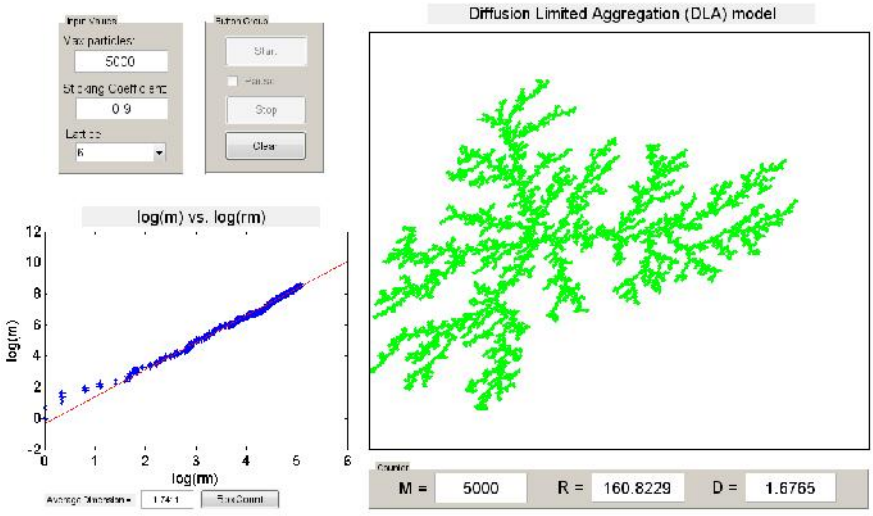


Figure 5.4(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.9 and 6 lattice site

In Figure 5.4(a) and (b), the simulation of single cluster fractal found in Chitosan-$AgNO_3$ films was done by choosing the input values; Max particles:5000, 6 lattice sites and Sticking coefficients:0.1 and 0.9 with the final radius, *R* achieved are 117.41 and 160.82 with fractal dimension value of 1.7872 and 1.6765 respectively. In both figures, the former has much denser branches compared to the latter caused by the lower value of sticking coefficient.

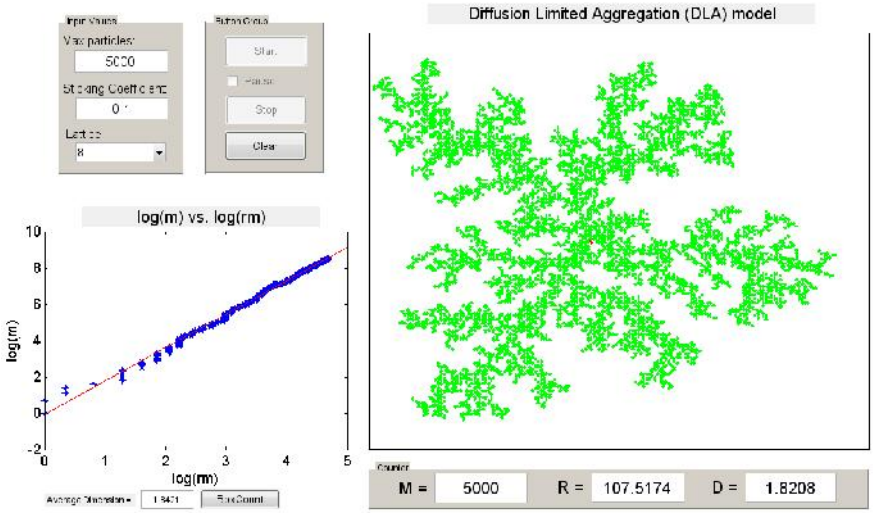


Figure 5.5(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.1 and 8 lattice site

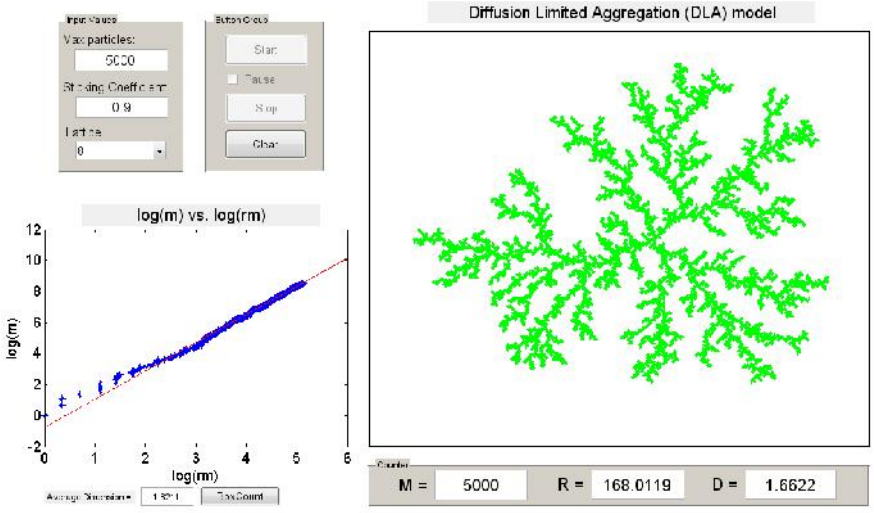


Figure 5.5(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.9 and 8 lattice site

Similar outcomes can be expected for the simulations in figure 5.5(a) and (b) where the simulation of single cluster fractal found in Chitosan-$AgNO_3$ films was done by choosing the input values; Max particles: 5000, 8 lattice sites and Sticking coefficients: 0.1 and 0.9 with the final radius, *R* achieved are 107.52 and 168.01 with fractal dimension value of 1.8208 and 1.6622 respectively.

The simulated single cluster fractal aggregates of the PEO-NH_4I film with combinations of values of number of particles, sticking coefficient and number of lattice sites are depicted in Figures 5.6-5.8 below.

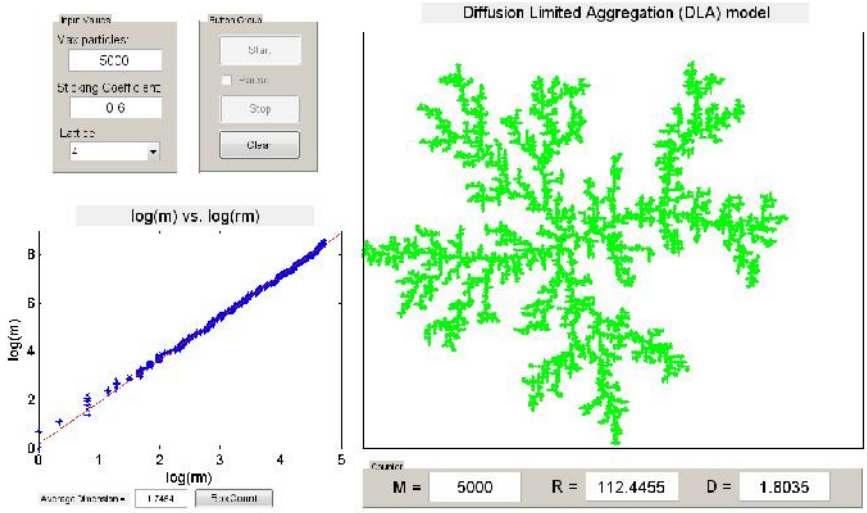


Figure 5.6(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.6 and 4 lattice site

In Figure 5.6(a) and (b), the simulation of single cluster fractal found in PEO-NH_4I films was done by choosing the input values; Max particles:5000, 4 lattice sites and Sticking coefficient:0.6 and 0.99 with the final radius, *R* achieved are 112.45 and 125.13 with fractal dimension value of 1.8035 and 1.7636 respectively. In both figures, the former has much denser branches compared to the latter caused by the lower value of sticking coefficient.

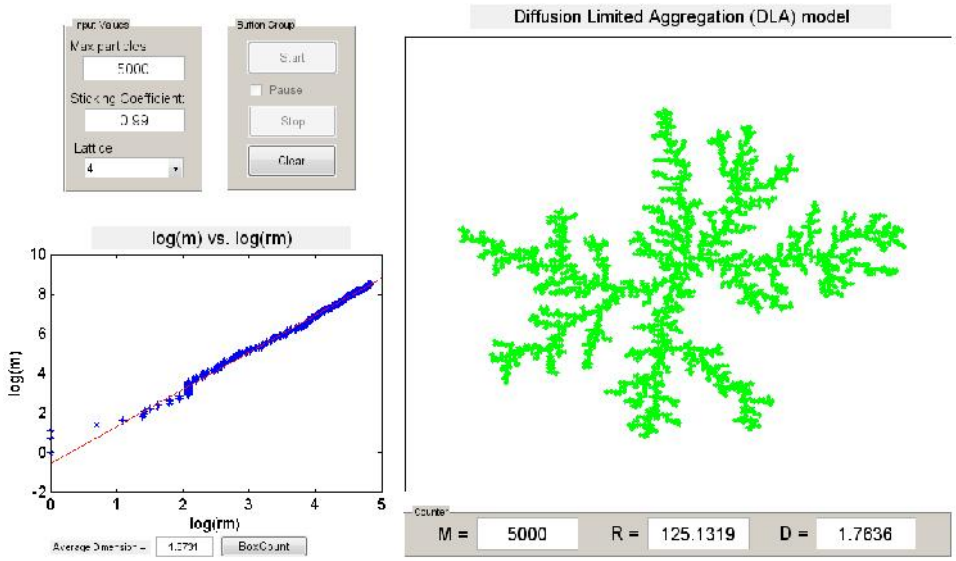


Figure 5.6(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.99 and 4 lattice site

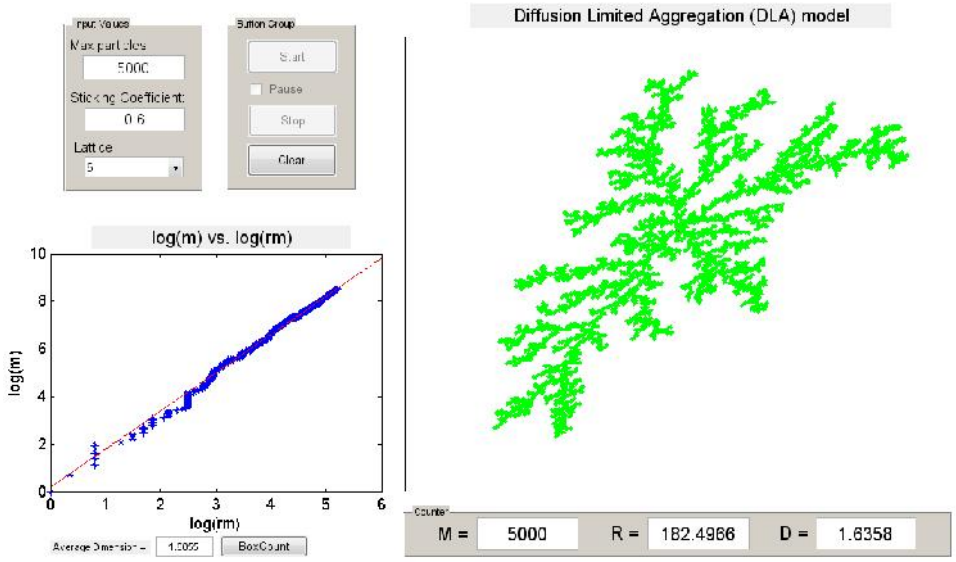


Figure 5.7(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.6 and 6 lattice site

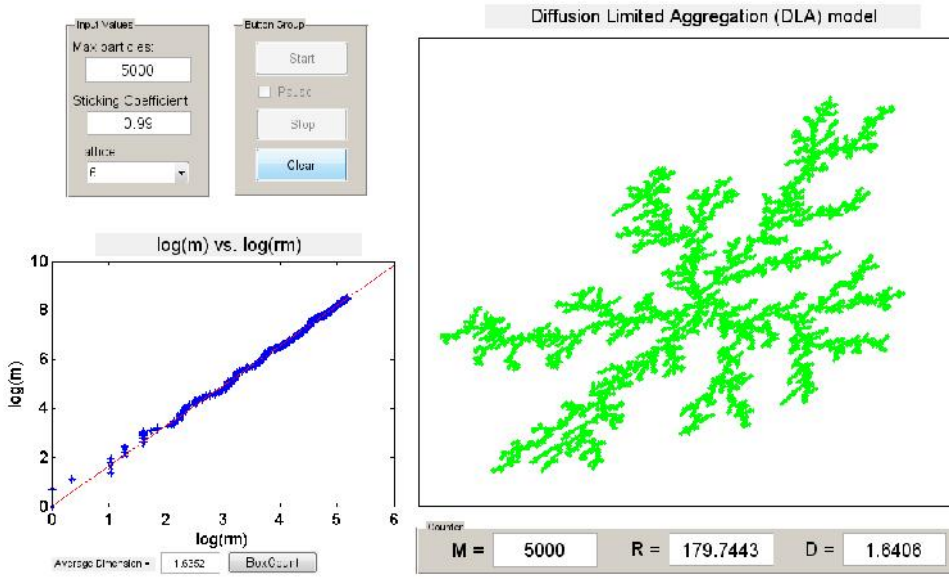


Figure 5.7(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.99 and 6 lattice site

In Figure 5.7(a) and (b), the simulation of single cluster fractal found in PEO-NH_4I films was done by choosing the input values; Max particles:5000, 6 lattice sites and Sticking coefficient:0.6 and 0.99 with the final radius, *R* achieved are 182.5 and 179.74 with fractal dimension value of 1.6358 and 1.6406 respectively. In both figures, the former has much denser branches compared to the latter caused by the lower value of sticking coefficient.

Similar outcomes can be expected for the simulations in figure 5.8(a) and (b) where the simulation of single cluster fractal found in PEO-NH_4I films was done by choosing the input values; Max particles: 5000, 8 lattice sites and Sticking coefficients: 0.6 and 0.99 with the final radius, *R* achieved are 144.46 and 165.13 with fractal dimension value of 1.7127 and 1.6678 respectively.

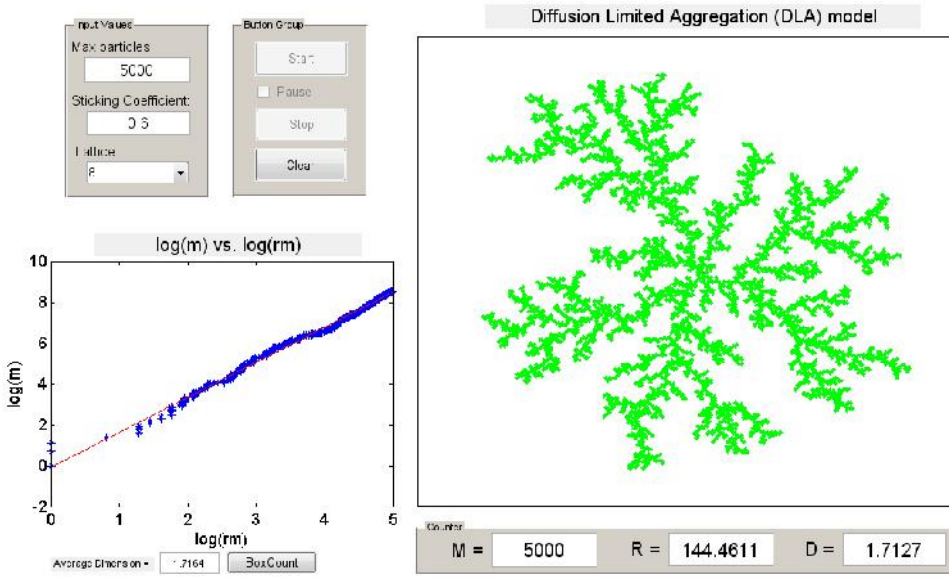


Figure 5.8(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.6 and 8 lattice site

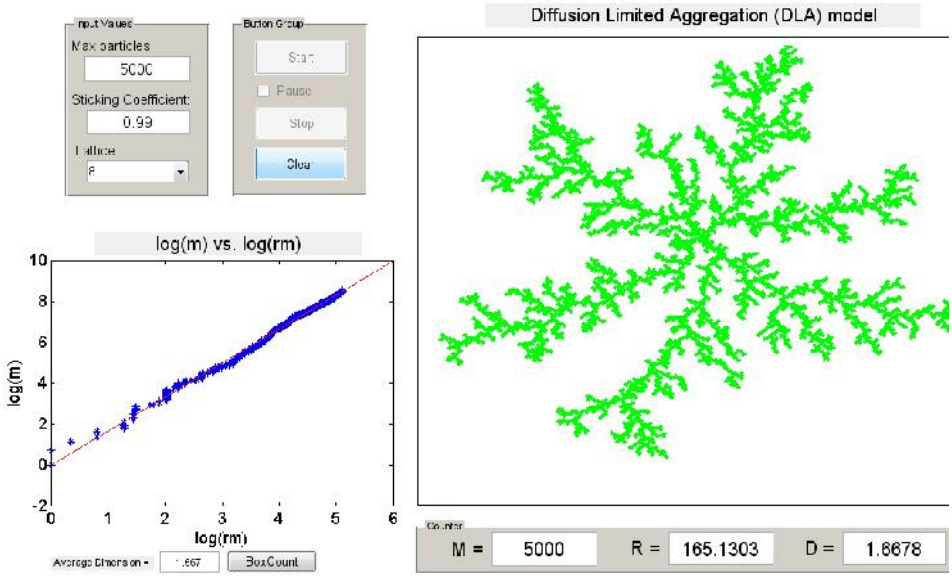


Figure 5.8(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.99 and 8 lattice site

For PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films, the simulated single cluster fractal aggregates with combinations of values of number of particles, sticking coefficient and number of lattice sites are depicted in Figures 5.9-5.11 below.

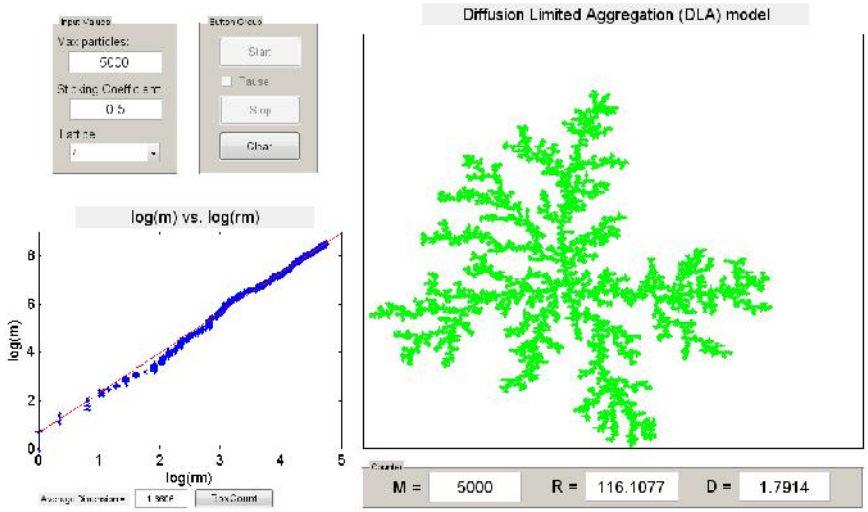


Figure 5.9(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.5 and 4 lattice site

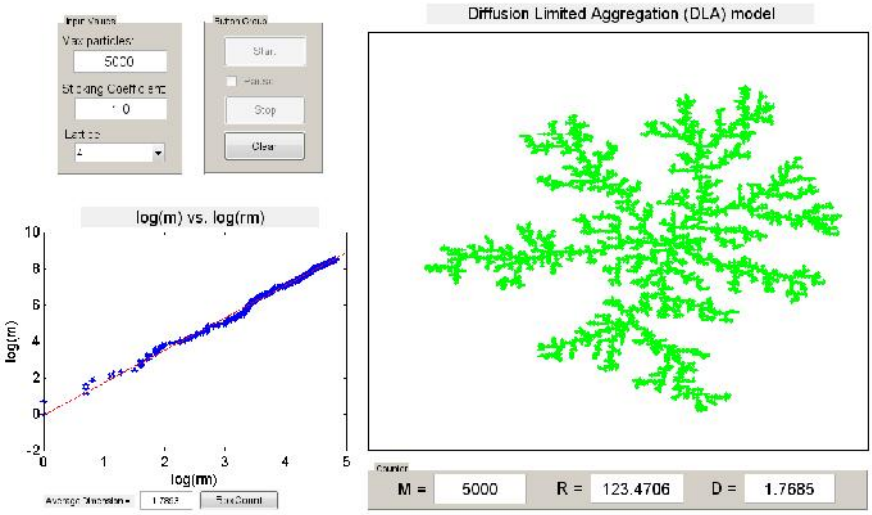


Figure 5.9(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 1.0 and 4 lattice site

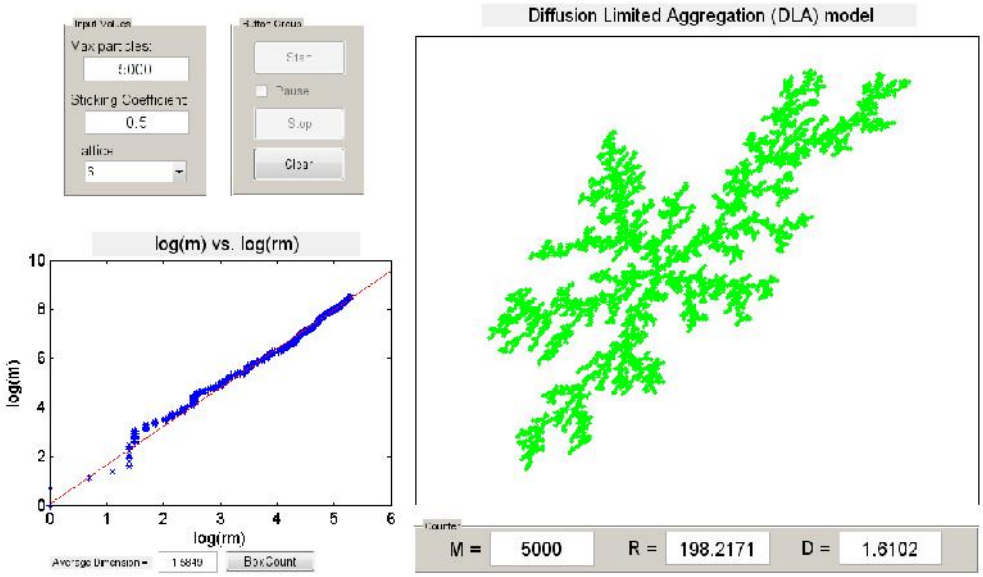


Figure 5.10(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.5 and 6 lattice site

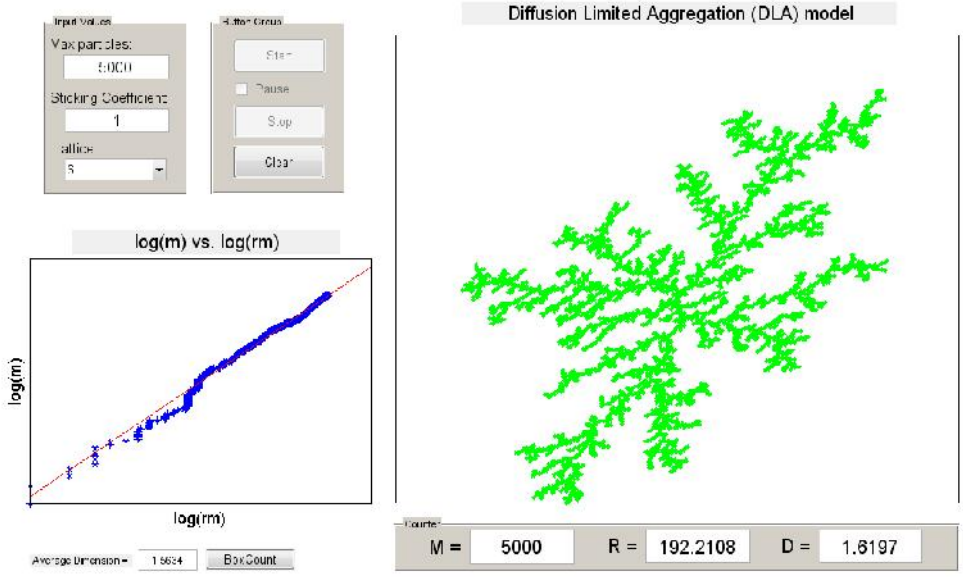


Figure 5.10(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 1.0 and 6 lattice site

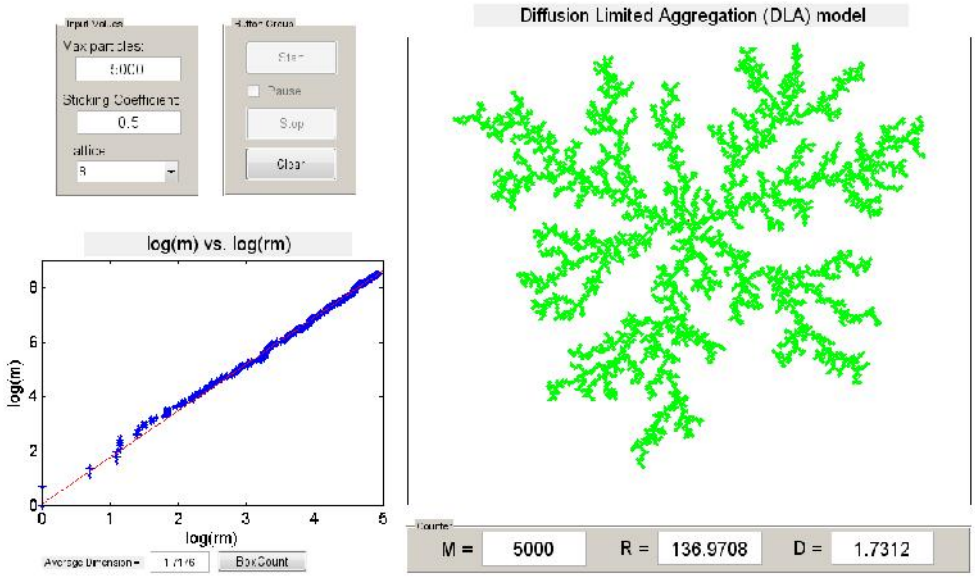


Figure 5.11(a) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 0.5 and 8 lattice site

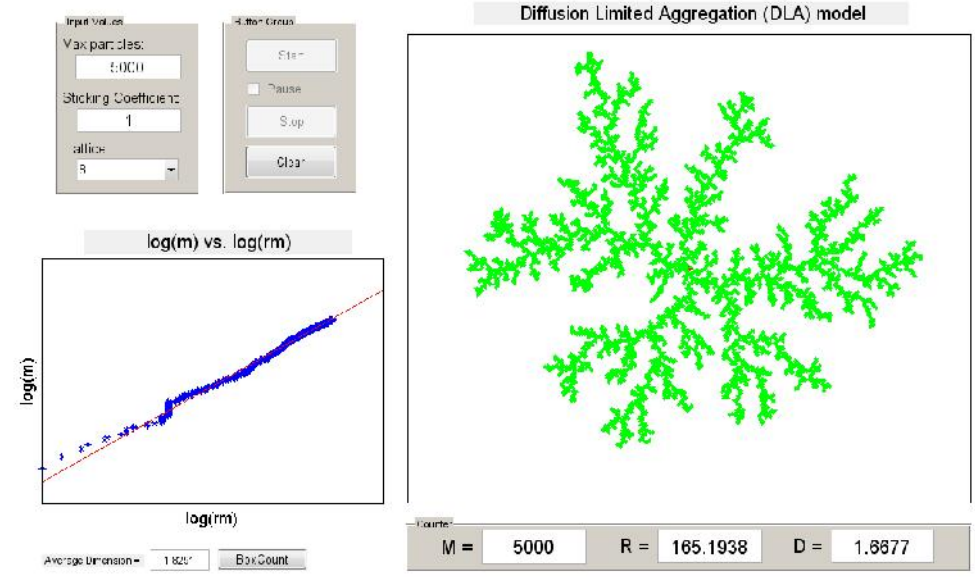


Figure 5.11(b) Single cluster fractal growth simulation with *M*=5000, sticking coefficient of 1.0 and 8 lattice site

According to the above results, most of the simulated single cluster fractal growth pattern follows a certain form in terms of shape and density. This can be regarded as a consistent result and can be perceived as a further improvement in the analysis of such similar fractal growth simulation as done in previously related study (Amir et al., 2010; 2011).

Improvements on the simulation work by the implementation of three fractal parameter values: max particles, *M*, sticking coefficient, α and lattice sites, *n*, has drastically contribute a lot to the success of developing computer programs for the generation of fractal patterns obtained in the chosen polymer electrolyte films. As *n* increases, the parameter values *M* and α can be adjusted accordingly to get a simulation image that is closely identical to the original fractal patterns observed in the polymer films. Tables 5.1 – 5.3 present the visual comparison of the images of the original fractal patterns and their simulated ones with the combination of all three fractal parameters. By comparing the original fractal patterns with the simulated fractals using different number of lattice sites, the simulated image that closely resembles the original fractal patterns can be identified.

Table 5.1: The comparison of original fractal patterns observed in the chitosan-$AgNO_3$ film with their simulated ones employing 3 simulation parameters

Original fractal patterns as shown in Figure 4.1	Simulated fractal patterns with 3 simulation parameters: max particles, M; sticking coefficient, α; numbers of lattice sites, *n*		
	n = 4	*n* = 6	*n* = 8
 4.1(a)	 *M* =500 *α* =1.0	 *M* =500 *α* =0.7	 *M* =500 *α* =0.5
 4.1(b)	 *M* =1500 *α* =0.5	 *M* =1500 *α* =0.7	 *M* =1500 *α* =1.0

Table 5.1 continued...

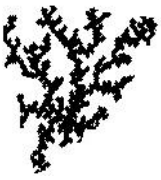
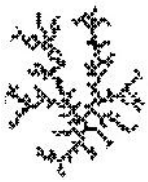
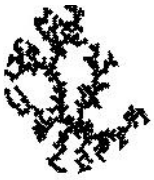
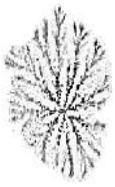
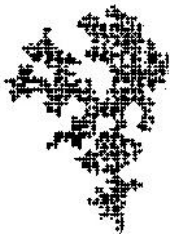
 4.1(c)	 M =2500 α =0.4	 M =2500 α =0.7	 M =2500 α =0.25
 4.1(d)	 M =2500 α =0.25	 M =2500 α =1.0	 M =2500 α =1.0
 4.1(e)	 M =3000 α =1.0	 M =3000 α =1.0	 M =3000 α =0.5
 4.1(f)	 M =5000 α =0.7	 M =5000 α =1.0	 M =5000 α =0.5
 4.1(g)	 M =5000 α =1.0	 M =5000 α =0.5	 M =5000 α =0.5
 4.1(h)	 M =5000 α =0.5	 M =5000 α =1.0	 M =5000 α =0.5

Table 5.1 continued...

 4.1(i)	 *M* =6000 α =0.75	 *M* =6000 α =1.0	 *M* =6000 α =1.0
 4.1(j)	 *M* =6500 α =0.25	 *M* =6500 α =0.1	 *M* =6500 α =0.25

By determining the shape of a cluster, branching and cluster density, the most accurate simulation image can be determined. Studying a comparison of one of the original fractal pattern from Figure 4.1(j) with its simulated patterns in Table 5.1, the closely resemble simulated pattern is the one with parameter values of n = 8, M= 6500 and α = 0.25. In the smaller size fractal pattern of Figure 4.1 (a) for example, the closest resemblance of the simulated patterns uses parameter values of n = 6, M= 500 and α = 0.7.

Table 5.2: The comparison of original fractal patterns observed in the PEO-NH_4I film with their simulated ones employing 3 simulation parameters

Original fractal patterns as shown in Figure 4.2	Simulated fractal patterns with 3 simulation parameters: max particles, M; sticking coefficient, α; numbers of lattice sites, *n*		
	n = 4	*n* = 6	*n* = 8
 4.2(a)	 *M* =500 α =0.6	 *M* =500 α =0.9	 *M* =500 α =0.1
 4.2(b)	 *M* =750 α =0.9	 *M* =750 α =0.1	 *M* =750 α =0.05

Table 5.2 continued...

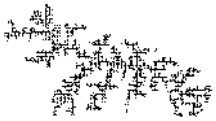
 4.2(c)	 M =1000 α = 0.5	 M =1000 α =0.7	 M =1000 α =0.5
 4.2(d)	 M =1500 α =0.4	 M =1500 α =0.25	 M =1500 α =0.05
 4.2(e)	 M = 2500 α = 0.55	 M =2500 α = 0.25	 M = 2500 α = 0.125
 4.2(f)	 M =2500 α =0.3	 M =2500 α =0.5	 M =2500 α =1.0
 4.2(g)	 M = 3000 α = 0.7	 M =3000 α =0.9	 M =3000 α =0.3
 4.2(h)	 M =5000 α =0.75	 M =5000 α =0.25	 M =5000 α =0.1
 4.2(i)	 M =5000 α =0.05	 M =5000 α =0.25	 M =5000 α =0.1

Table 5.2 continued...

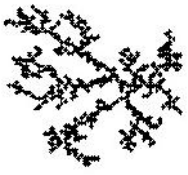
 4.2(j)	 *M* =5000 α =0.35	 *M* =5000 α =0.25	 *M* =5000 α =0.6

Taking into account the different parameters involved in the comparison of the original fractal pattern from Figure 4.2(a) with its simulated patterns in Table 5.2, the closely resemble simulated pattern is the one with parameter values of n = 8, M= 500 and α = 0.1. For bigger size fractal pattern of Figure 4.2 (j), the closest resemblance of the simulated pattern applies parameter values of n = 4, M= 5000 and α = 0.35.

Table 5.3: The comparison of original fractal patterns observed in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film with their simulated ones employing 3 simulation parameters

Original fractal patterns as shown in Figure 4.3	Simulated fractal patterns with 3 simulation parameters: max particles, M; sticking coefficient, α; numbers of lattice sites, *n*		
	n = 4	n = 6	n = 8
 4.3(a)	 *M* =500 α =0.5	 *M* =500 α =0.25	 *M* =500 α =0.5
 4.3(b)	 *M* =1000 α =0.5	 *M* =1000 α =0.25	 *M* =1000 α =0.3
 4.3(c)	 *M* =1000 α =0.4	 *M* =1000 α =0.25	 *M* =1000 α =1.0

Table 5.3 continued...

 4.3(d)	 M =1000 α =0.1	 M =1000 α =0.9	 M =1000 α =1.0
 4.3(e)	 M =1000 α =0.6	 M =1000 α =1.0	 M =1000 α =1.0
 4.3(f)	 M =2500 α =0.6	 M =2500 α =0.25	 M =2500 α =0.09
 4.3(g)	 M =2500 α =0.6	 M =2500 α =0.75	 M =2500 α =0.09
 4.3(h)	 M =2500 α =0.6	 M =2500 α =1.0	 M =2500 α =0.9
 4.3(i)	 M =5000 α =0.5	 M =5000 α =0.8	 M =5000 α =1.0
 4.3(j)	 M =5000 α =1.0	 M =5000 α =0.5	 M =5000 α =0.5

As for the comparison of one of the original fractal pattern from Figure 4.3(a) with its simulated patterns in Table 5.3, the nearest similarity between original pattern and simulated pattern is the one with parameter values of n = 8, M= 500 and α = 0.5. While for larger size fractal pattern of Figure 4.3 (j), the closest resemblance of the simulated patterns applies parameter values of n = 6, M= 5000 and α = 0.5.

Looking at the tables, the most effective and suitable parameter values for different types of fractals observed in all three polymer films can be ascertained. For chitosan-$AgNO_3$ film, the parameter values of n = 4 or n = 6 with variation of smaller number of M and 0.4 < α <0.7 are defining factors in achieving the best fit simulation image for smaller size original fractal patterns (Figure 4.1(a)-(c)). Range of parameter values for simulation of larger fractals as seen in Figures 4.1(d)-(j) are found to be n = 4 or n = 8 with higher values of M and 0.25 < α <1.0. In the simulation of single cluster fractals of PEO-NH_4I film, the parameter values of n = 8 with variation of smaller number of M and 0.05 < α <0.5 are found to be the best suitable simulation image for smaller size original fractal patterns (Figure 4.2(a)-(c)). Range of parameter values for simulation of larger fractals as seen in Figures 4.2(d)-(j) are found to be n = 4 or n = 8 with higher values of M and 0.05 < α <0.35. Whilst for PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film, the parameter values of n = 4 with variation of smaller number of M and 0.1 < α <0.6 are defining factors in achieving the best fit simulation image for smaller size original fractal patterns (Figure 4.3(a)-(e)). Range of parameter values for simulation of larger fractals as seen in Figures 4.3(f)-(j) are found to be n = 4 or n = 8 with higher values of M and 0.09 < α < 1.0.

5.2 Simulation Program Validation

For a single cluster, when the simulation of the aggregate was done on a square lattice (Figure 5.12), the figure showed that there were four branches from the center when the aggregate was produced on a square lattice with a sticking coefficient of 1.0. The fractal dimension for this aggregate is 1.6587. This is comparable to the value of 1.66, found in a paper (Witten and Sander, 1981) by Witten and Sander.

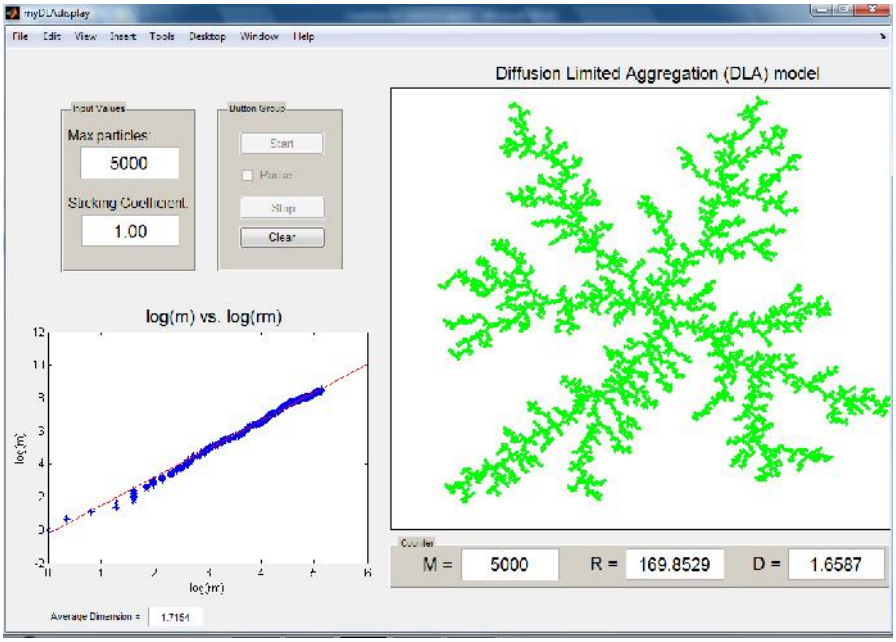


Figure 5.12 : Aggregate on a square lattice with a sticking coefficient of 1.0

The fractal dimension of the aggregate grown with the sticking coefficient of 0.5 is 1.7022 (Figure 5.13) even though the aggregate looks denser than the one with sticking coefficient of 1.0.

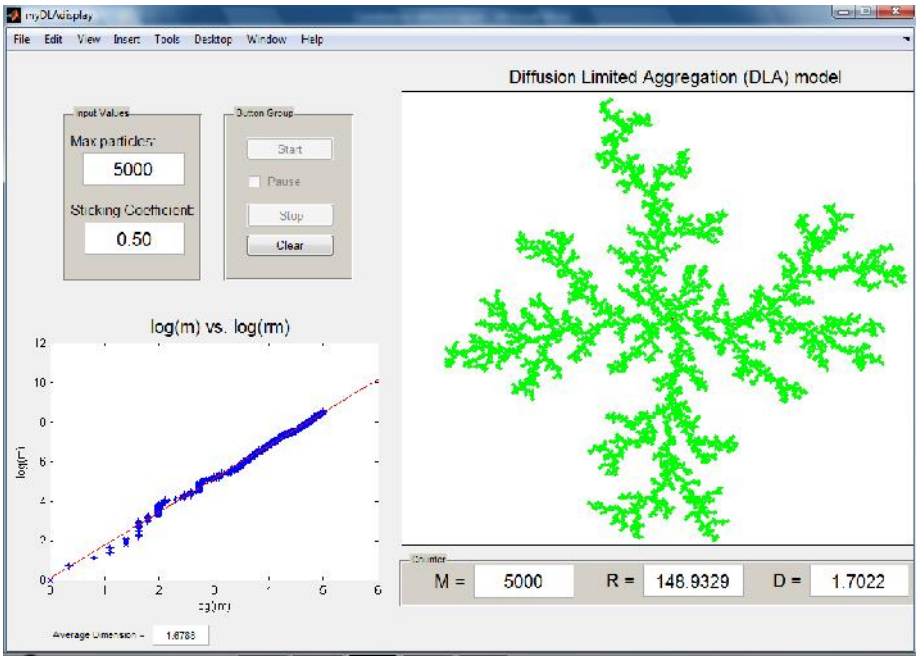


Figure 5.13 : Aggregate on a square lattice with a sticking coefficient of 0.5

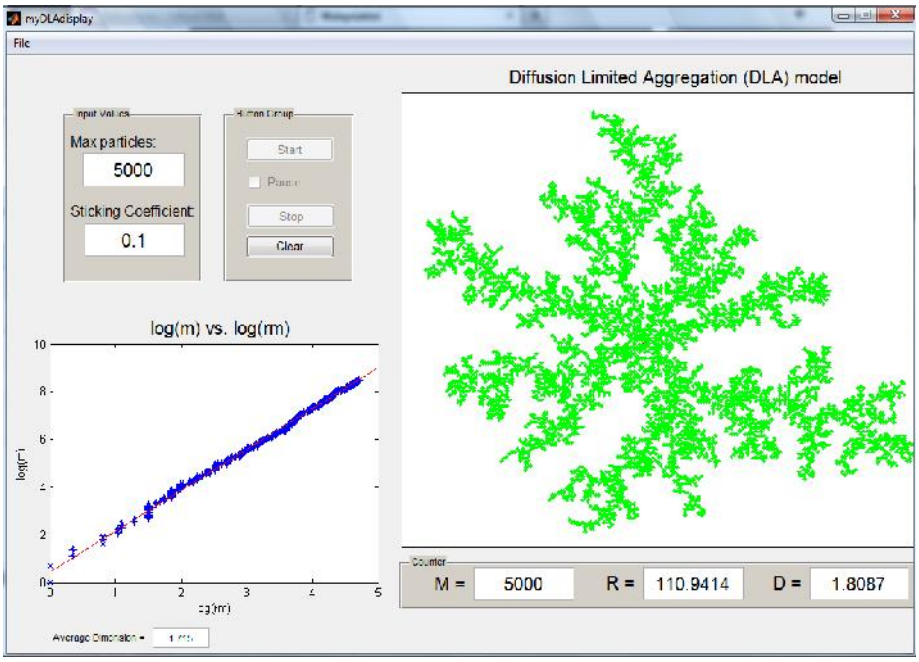


Figure 5.14 : Aggregate on a square lattice with a sticking coefficient of 0.1

The features become even thicker as the sticking coefficient is decreased to 0.1 (Figure 5.14). The trend continues until the features merge together and the aggregate becomes one big clump. The fractal dimension of the aggregate with a sticking coefficient of 0.1 is 1.8087.

Furthermore, a couple of similar fractal patterns found in bacterial growth, electrodeposition and mineralization were also utilized for simulation. The results were interesting and can further verify the usefulness of the simulation program developed in this research work. Figure 5.15 shows a simulation of an image of original fractal pattern (inset) found in bacterial growth (Matsushita et al., 2004) using the developed simulation program.

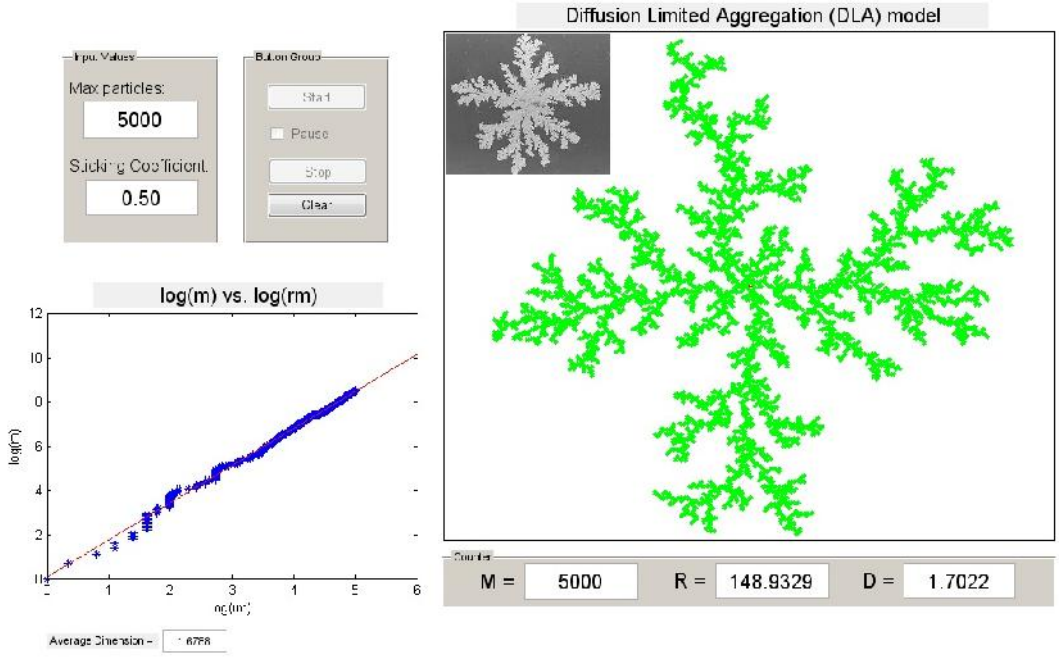


Figure 5.15 : Simulation of an image of original fractal pattern (inset) found in bacterial growth using the developed simulation program

Matsushita et al. (2004) reported that the pattern shown in Figure 5.15 (inset) is self-similar fractal and obtained the fractal dimension D = 1.72 ± 0.02, which is in good agreement with that of two-dimensional DLA cluster. The simulation employing max particles of 5000 particles with sticking coefficient of 0.5 manages to produce DLA cluster of D = 1.7022 . It is evident from Figure 5.15, that the simulation looks very similar to the original fractal pattern of bacterial growth (inset).

While Shaikh et al. (2009) have also done similar simulation study of electrodeposition growth of fractal using $CuSO_4$ solution with D = 1.63 as exhibited in the inset of Figure 5.16. From the figure, it is clearly visible that the simulation with D = 1.68 looks similar to its original electrodeposition fractal image (inset).

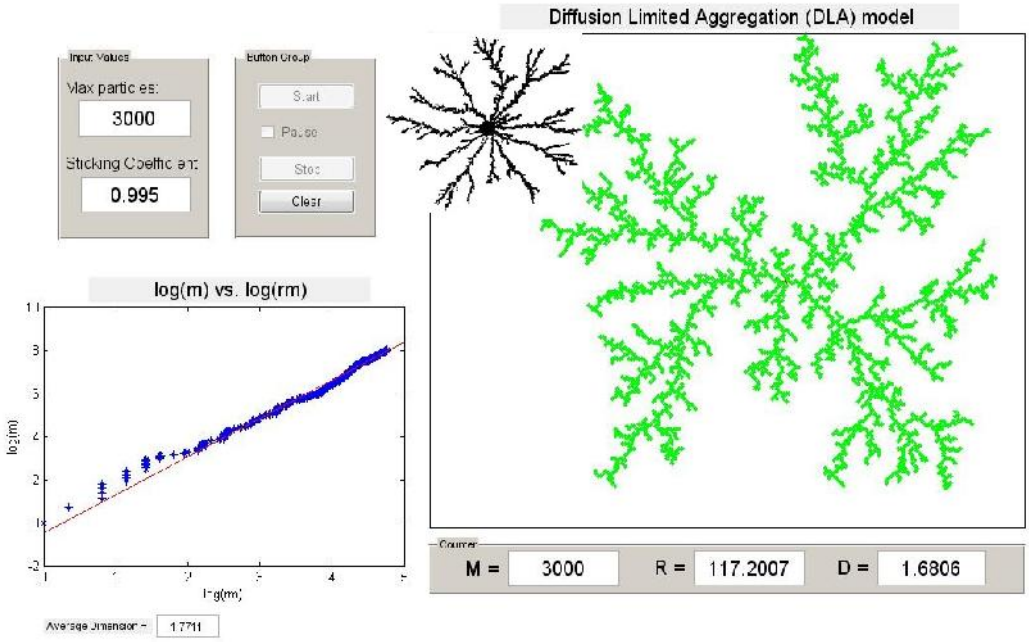


Figure 5.16 : Simulation of an image of original fractal pattern (inset) found in electrodeposition growth of fractal using $CuSO_4$ solution at higher voltage condition (12 V) using the developed simulation program

Obviously from the two simulation samples of similar fractal growth observed in other previous works have proven that the simulation program utilized in the current study succeeds in coming up with neat results.

5.3 Results of Overall Fractal Growth Analysis

With the results of the simulations of the single cluster fractal growth as discussed in the two previous sections, it is comprehensible to compare the simulation results with the cultured fractals. From the comparisons, further understanding of the morphology of any single or multiple cluster fractal growth patterns can be achieved. Tables 5.4-5.6 typify the simulation and cultured fractal growth patterns of all the three types of polymer electrolyte films: chitosan-$AgNO_3$; PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 with comparison of their fractal dimension values.

Table 5.4: Fractal dimension values for the simulated and experimentally cultured fractals of chitosan-$AgNO_3$ film

Comparison results		
Simulated		Cultured
M=500 α = 0.7 *n* =6	 1.713 ± 0.044	 4.1(a) 1.705 ± 0.035
M=1500 α =0.5 *n* =4	 1.714 ± 0.039	 4.1(b) 1.714 ± 0.044

Table 5.4 continued...

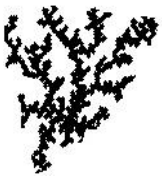
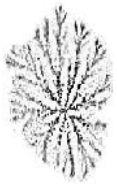
M=2500 α = 0.4 *n* =4	 1.715 ± 0.047	 4.1(c) 1.726 ± 0.049
M=2500 α =1.0 *n* =6	 1.724 ± 0.042	 4.1(d) 1.707 ± 0.053
M=3000 α = 0.5 *n* =8	 1.702 ± 0.046	 4.1(e) 1.698 ± 0.036
M=5000 α = 1.0 *n* =6	 1.714 ± 0.039	 4.1(f) 1.709 ± 0.045
M=5000 α = 0.5 *n* =8	 1.731 ± 0.037	 4.1(g) 1.732 ± 0.039
M=5000 α = 1.0 *n* =6	 1.728 ± 0.045	 4.1(h) 1.729 ± 0.043

Table 5.4 continued...

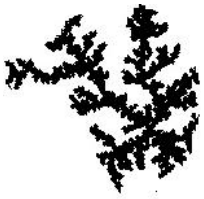
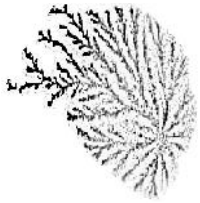
M=6000 α = 1.0 *n* =6	 1.765 ± 0.046	 4.1(i) 1.774 ± 0.037
M=6500 α =0.25 *n* =4	 1.766 ± 0.038	 4.1(j) 1.761 ± 0.015

Table 5.5: Fractal dimension values for the simulated and experimentally cultured fractals of PEO-NH_4I film

Comparison results		
Simulated		Cultured
M=500 α = 0.1 *n* =8	 1.699 ± 0.055	 4.2(a) 1.709 ± 0.053
M=750 α = 0.05 *n* =8	 1.734 ± 0.043	 4.2(b) 1.728 ± 0.031
M=1000 α = 0.5 *n* =8	 1.715 ± 0.047	 4.2(c) 1.714 ± 0.015
M=1500 α = 0.05 *n* =8	 1.757 ± 0.042	 4.2(d) 1.753 ± 0.045

Table 5.5 continued...

M=2500 α = 0.125 *n* =8	 1.754 ± 0.048	 4.2(e) 1.768 ± 0.047
M=2500 α = 0.3 *n* =4	 1.705 ± 0.053	 4.2(f) 1.741 ± 0.039
M = 3000 α = 0.3 *n* =8	 1.724 ± 0.042	 4.2(g) 1.714 ± 0.043
M = 5000 α = 0.25 *n* =6	 1.766 ± 0.042	 4.2(h) 1.746 ± 0.034
M = 5000 α = 0.1 *n* =8	 1.733 ± 0.047	 4.2(i) 1.752 ± 0.046
M = 5000 α = 0.35 *n* =4	 1.785± 0.045	 4.2(j) 1.794 ± 0.043

Table 5.6: Fractal dimension values for the simulated and experimentally cultured fractals of PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film

Comparison results		
Simulated		Cultured
M = 500 α = 0.5 n =8	 1.683 ± 0.043	 4.3(a) 1.667 ± 0.055
M =1000 α = 0.25 n =6	 1.721 ± 0.042	 4.3(b) 1.718 ± 0.045
M =1000 α = 0.4 n =4	 1.724 ± 0.045	 4.3(c) 1.706 ± 0.043
M =1000 α = 0.1 n =4	 1.747 ± 0.044	 4.3(d) 1.741 ± 0.049
M =1000 α = 0.6 n =4	 1.707 ± 0.041	 4.3(e) 1.713 ± 0.041
M =2500 α = 0.09 n =8	 1.738 ± 0.045	 4.3(f) 1.731 ± 0.042

Table 5.6 continued...

M =2500 α = 0.09 n =8	 1.751 ± 0.045	 4.3(g) 1.749 ± 0.028
M =2500 α = 0.9 n =8	 1.749 ± 0.045	 4.3(h) 1.743 ± 0.023
M =3000 α = 0.5 n =4	 1.755 ± 0.049	 4.3(i) 1.757 ± 0.012
M =5000 α = 1.0 n =4	 1.778 ± 0.053	 4.3(j) 1.786 ± 0.044

The tables show that the fractal dimension values of the simulated fractals are comparable with the fractal dimension values obtained from their respective experimentally cultured ones. For instance, the fractal dimension of the experimentally cultured fractal in Figures 4.1(j), 4.2 (j) and 4.3(j) are 1.761 ± 0.015, 1.794 ± 0.043 and 1.786 ± 0.044, while the fractal dimension values for their simulated patterns are 1.766 ± 0.038, 1.785± 0.045 and 1.778 ± 0.053 respectively.

From the tables, it is clear that the percentage difference of fractal dimension values between the experimental and simulated patterns are marginally close. The error values for both experimentally and simulated patterns are found to be so small with standard deviation of less than 3%. These show that the simulated fractal patterns are of fairly good

conformity with the fractal patterns observed in the chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 polymer films.

For every type of cultured fractals, the size, shape and density of particles were found to be different. Thus by implementing simulation of different parameters facilitates more efficient way of getting better comparison between the simulated and cultured fractal growth pattern. By making this comparison, the understanding on how such fractal growth patterns occur can further be enhanced. It can be averred that for smaller size of any single cluster, smaller values of max particles should be applied in the simulation. Generally, shape of a cluster with more lattice sites tends to give a more compact look and the lower sticking coefficient presents a denser appearance.

C H A P T E R 6

RESULTS AND DISCUSSIONS OF SIMULATED MULTIPLE CLUSTER FRACTALS

Simulation works on multicluster type of fractal growth patterns were carried out by applying certain rules for each type of simulation. Further discussions on the rules and conditions applied in each type of simulation can be found in the following sections.

6.1 Simulation Program of Multiple Cluster Fractal Growth Pattern

In the simulation of multicluster fractal growth pattern, one important feature is the ability to run fractal growth of multiple seeds by implementing nucleation centers (seed) randomly within an area with a specific radius, *R* that can accommodate large number of particles, *M*. The aim of this simulation is to replicate the growth of multicluster fractal growth pattern observed in the cultured fractals as seen in Chapter 4. For that purpose, a simulation program of multicluster fractal growth pattern has been successfully developed and the results are found to be quite interesting to study.

Referring to figures of the cultured fractals of chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films in Chapter 4, there are multicluster fractal growth patterns that form within areas where the number of clusters depends on the number of nucleation sites (seeds). From close observation, in these areas, it appears that they consist of big and small clusters together in a group of multiple clusters. For chitosan-$AgNO_3$ (Figure 4.1), the total number of clusters is around 30. In Figure 4.2, the PEO-NH_4I film has around 20 clusters and the PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film (Figure 4.3)

has about 15 visible clusters. Most of the clusters within the films are found to be scattered and do not overlap each other.

With this consideration, the simulation of multicluster growth pattern was done by varying the number of seeds representing clusters within certain areas. Figure 6.1 demonstrates a simulation of an area of multicuster fractal growth pattern observed in the cultured chitosan-$AgNO_3$ polymer electrolyte film with 3 seeds.

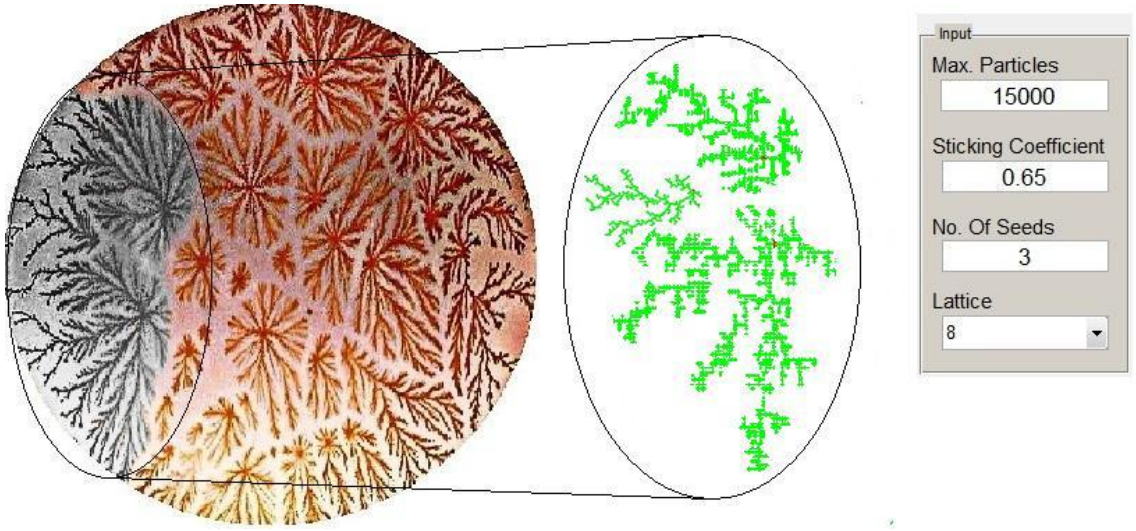


Figure 6.1: Simulation of an area of multicluster fractal growth pattern observed in the cultured chitosan-$AgNO_3$ polymer electrolyte film with 3 seeds

The simulation of multicluster fractal growth pattern shown in Figure 6.1 clearly resembles the group of clusters observed in the chitosan-$AgNO_3$ polymer electrolyte film that is shown in Figure 4.9(c). In both the simulated and original fractal growth pattern, neighboring clusters tend to grow away from each other and do not overlap. Clusters grow in various sizes and in outward direction from their nucleation sites. The parameters chosen in this simulation are M = 15000 particles, sticking coefficient of 0.65 and 8 lattice

sites. Simulation using different parameter values is perhaps more suitable for other types of original fractals depending on the form and shape.

As for the simulation of the multicluster fractal growth patterns in the PEO-NH_4I film, different parameter values were applied. The parameters are M = 10000 particles, sticking coefficient of 0.25 and 8 lattice sites. Figure 6.2 shows the simulation of part of the group of multicluster fractals with 3 nucleation sites as seen in the film of PEO-NH_4I depicted in Figure 4.10(c).

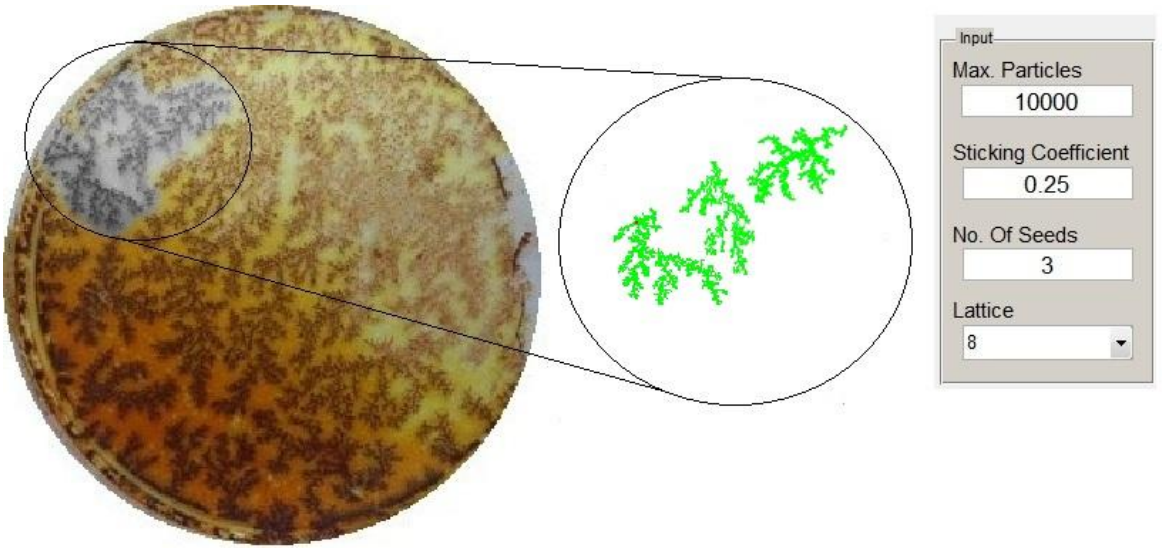


Figure 6.2: Simulation of an area (in grey) of multicluster fractal growth pattern observed in the cultured PEO-NH_4I polymer electrolyte film with 3 seeds

In the simulation of the multicluster fractal growth patterns in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 film, the parameter values are M = 15000 particles, sticking coefficient of 0.9 and 4 lattice sites. Figure 6.3 shows the simulation of part of the group of multicluster fractals with 4 nucleation sites as seen in the PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 membrane in Figure 4.11(a).

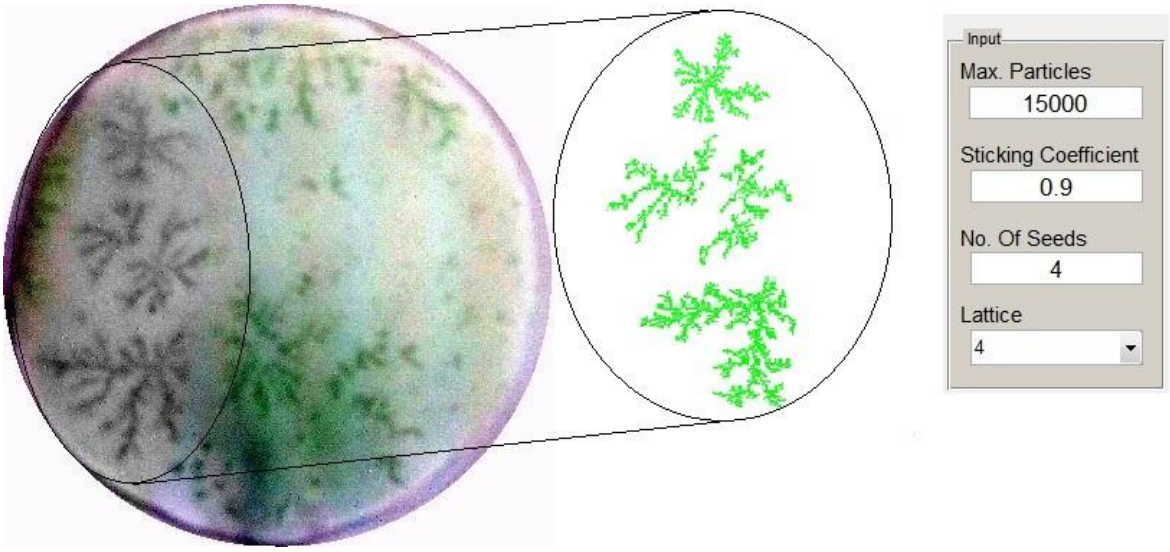


Figure 6.3: Simulation of an area (in grey) of multicluster fractal growth pattern observed in the cultured PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 polymer electrolyte film with 4 seeds

From all the cultured fractals, the multicluster fractals of PEO-NH_4I film have denser clusters compared to the chitosan-$AgNO_3$ and PVDF-HFP/PEMA-$NH_4CF_3SO_3$-Cr_2O_3 films. Thus the parameter values were set with higher values of max particles and lower sticking coefficient with 8 lattice sites. For the multicluster simulations shown in Figs. 6.1- 6.3, fractal dimension values of each of the simulated fractals in the selected clusters and its corresponding original patterns are tabulated in Tables 6.1- 6.3 respectively. The tables clearly show for each simulated cluster, the fractal dimension values are almost identical and are in conformity with the original patterns found in the polymer films.

Table 6.1: Fractal dimension values of each of the simulated fractal patterns for the multicluster simulation as shown in Figure 6.1

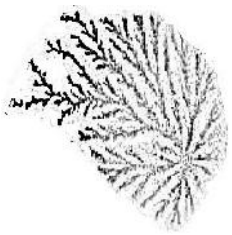
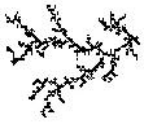
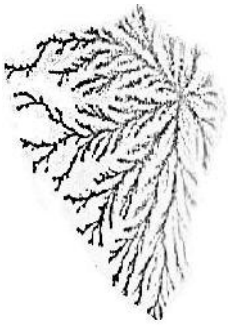
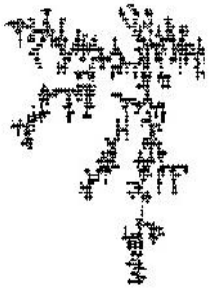
Experimentally cultured	Simulated
 $f_D = 1.774 \pm 0.037$	 $f_D = 1.773 \pm 0.041$
 $f_D = 1.698 \pm 0.036$	 $f_D = 1.707 \pm 0.042$
 $f_D = 1.761 \pm 0.015$	 $f_D = 1.778 \pm 0.046$

Table 6.2: Fractal dimension values of each of the simulated fractal patterns for the multicluster simulation as shown in Figure 6.2

Experimentally cultured	Simulated
 $f_D = 1.707 \pm 0.044$	 $f_D = 1.713 \pm 0.044$
 $f_D = 1.726 \pm 0.042$	 $f_D = 1.764 \pm 0.048$
 $f_D = 1.768 \pm 0.047$	 $f_D = 1.756 \pm 0.042$

Table 6.3: Fractal dimension values of each of the simulated fractal patterns for the multicluster simulation as shown in Figure 6.3

Experimentally cultured	Simulated
 $f_D = 1.749 \pm 0.028$	 $f_D = 1.746 \pm 0.045$
 $f_D = 1.743 \pm 0.023$	 $f_D = 1.724 \pm 0.043$
 $f_D = 1.757 \pm 0.012$	 $f_D = 1.749 \pm 0.046$
 $f_D = 1.786 \pm 0.044$	 $f_D = 1.777 \pm 0.054$

Even though parameter values are set the same every time a simulation is run, different configuration of patterns were formed. This is due to the random movement of particles during simulation with any kind of fractal parameter values. Thus to obtain an almost exact simulation pattern compared to its original pattern, this requires repetition of

simulation program execution. Depending on the adeptness of obtaining satisfying result, number of repetitions may reach to as many as 20 repetitions.

6.2 Simulation Program Validation

To further verify the usefulness of the simulation program developed in this research work, a simulation of an image of original multicluster fractal pattern was done (Figure 6.4). The multicluster fractal pattern was obtained from electrochemical deposits grown on two cathodes (Bankar et al., 2007).

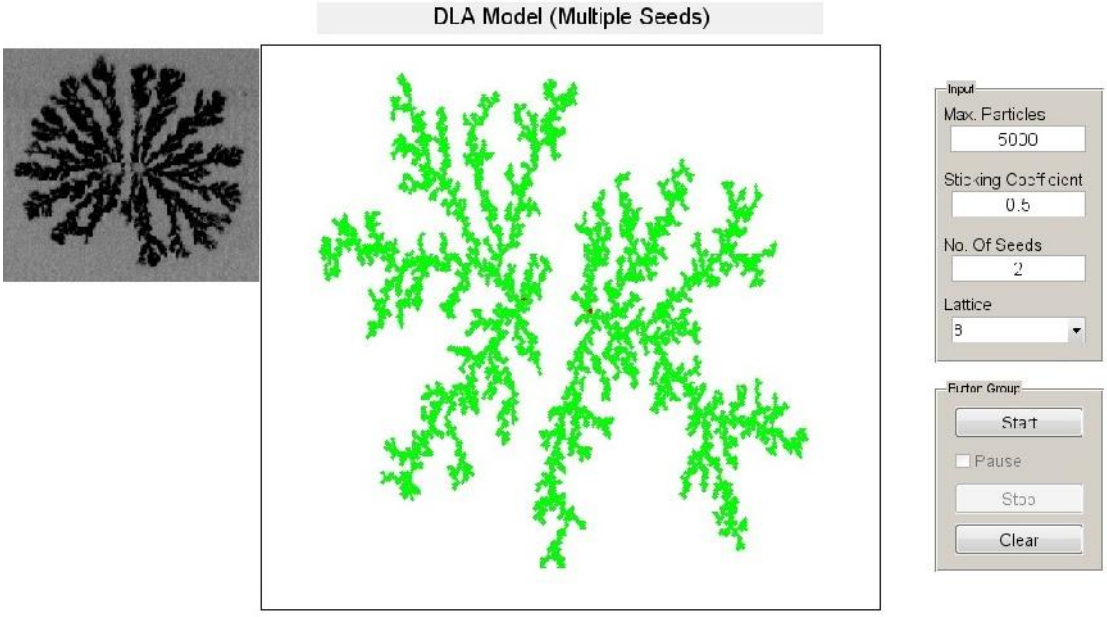


Figure 6.4: Simulation of multicluster fractal pattern of electrochemical deposits grown on two cathodes using the simulation program of multiple cluster fractal patterns with 2 seeds

Bankar et al. (2007) reported that the patterns obtained from electrochemical deposits grown on two cathodes as shown in Figure 6.4 (inset) are Zn electrodeposits using aqueous $ZnSO_4$ solution for two-cathodes producing segregated clusters and obtained the combined fractal dimension D =1.86 ± 0.01. Correspondingly, the simulation employing max particles of 5000 particles with sticking coefficient of 0.5 and 2 seeds with 8 lattice sites manages to produce DLA cluster that is almost similar to the reported D value of the original multicluster fractal pattern, with combined D = 1.8072. This shows that the simulation of a multicluster pattern of 2 seeds done in this work successfully produces result of good conformity compared to the original multicluster fractal patterns.

C H A P T E R 7

CONCLUSION AND SUGGESTION FOR FUTURE WORKS

The research described in this thesis contributes to the area in the domain of simulation of fractal patterns. The fractal patterns used for the simulation were obtained through experimental work. The experimental and simulation works explore aspects of both physical and mathematical perspectives. The processes of culturing fractals in the laboratory were carefully executed in order to get suitable images of fractal patterns for simulation purposes. As described in Chapter 4, the culturing of fractals were done using three types of polymer electrolyte films: chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3. From the results and discussion in the previous chapters, the fractal patterns formed were recognized as DLA type fractals.

7.1 Conclusion

The growth of fractals observed in this study is due to the random motion of aggregating species which are believed to be a triggering factor for a formation of any fractal pattern. To the best of the author's knowledge, research works on fractals were done only on laboratory experiments, theoretical modeling and experimental studies, or modeling and computer simulations. In the present work, integration of all the three approaches; experimental, modeling and simulation has successfully been implemented. The culturing of fractals was done using chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3 polymer electrolyte films, all prepared via solution casting technique. Considering single and multicluster fractals that formed in these polymer

electrolyte films, simulation programs for both single and multicluster fractals were designed and developed. To study the growth of these fractal patterns the simulation has been carried out based on the DLA model. The simulation techniques that were chosen have the capability of producing images with fractal dimension values comparable with those obtained for experimentally cultured fractals. This indicates that the computer program undertaking the DLA model has successfully come out with outputs that are in accordance with the original patterns found in the polymer electrolyte films. The success in growing fractals using the polymer films shows that besides applications in electrochemical devices, polymer electrolyte films are also suitable for the study of fractals.

Looking at the simulation results of the single cluster fractal patterns, they show remarkable resemblance to the original DLA type fractals cultured in the laboratory. In the simulation work, the lattice geometry is varied according to three number of lattice sites of 4, 6 and 8. Each lattice site produced an artifact of the model with an overall shape of the cluster related to the shape of the lattice. The clusters showed significant properties such as branching and screening, scale invariance and stochastic self-similarity. In the generation of a single cluster fractal, the size, shape and density can be set by varying the parameters of the number of particles, M, sticking coefficient, α and number of lattice sites, n. Alternatively, a way to accommodate the number of branching for the simulation of the single cluster fractals is by introducing certain production rules in the generation of the fractal growth. This has been done in the simulation using L-System approach as reported by Amir et al (2008). In this work, for example, adjustment to the number of branching for highly branched fractal patterns is tackled by applying fractal parameter of number of lattice site, $n = 8$ with combination of appropriate M and α values.

It is percipient that for chitosan-$AgNO_3$, PEO-NH_4I and PVDF-HFP/ PEMA-$NH_4CF_3SO_3$-Cr_2O_3 polymer electrolyte films, the simulation parameters can be adapted according to the main features of the fractal patterns observed in each film. For instance the fractal patterns noticed in the chitosan-$AgNO_3$ film consist of clusters of different sizes and shapes with bigger clusters and denser outlook. Thus for simulation of the patterns, the different in size can be attributed by implementing higher number of particles, M for bigger clusters and for denser fractals the sticking coefficient, α is set for lower values.

As for the shape of the fractal patterns, applying different number of lattice sites provides a wider range of options in representing the various shapes of every cluster. For example applying 6 lattice site produces a look of skewer shape since the particles move and stick in triangular direction. While with 4 lattice sites the shape of the cluster seems to produce squarer shape compared to applying 8 lattice sites where the particles move and stick not just in cross direction but also diagonally. Clearly the introduction of all the parameters in the simulation work of single cluster fractal patterns has successfully improved the way to simulate the many sort of fractal patterns observed in different types of polymer electrolyte films.

It is not adequate to just simulate a single cluster fractal pattern without considering that in most of the fractals observed in the polymer electrolyte films, there exist multiple clusters (multicluster) fractal patterns that do not overlap each other. It appears that the clusters are separated from each other by a certain boundary. Every cluster grows in each spot with particular nucleation site that accumulates into aggregates to form any single cluster that avoids contact with its neighboring cluster. In all the three types of polymer electrolyte films investigated in this thesis, the clusters do not touch each other and form

new cluster as occurred in cluster-cluster type aggregation. With that in mind, the simulation applying DLA model was mooted to be the best model for simulation of the multicluster fractal patterns.

Again in the simulation of multicluster fractal patterns, the simulation employs simulation parameters that were also implemented in the simulation of single cluster fractal patterns discussed earlier. For all the three types of polymer electrolyte films, the number of clusters ranges around 20-30 clusters of different sizes and shape. Therefore to simulate the multicluster fractal patterns, every neighboring cluster are divided by different regions containing 2 or more clusters that grows very close to each other but do not overlap each other. The multicluster simulation program developed in this work has successfully simulated the multicluster fractal patterns examined in all the three types of polymer electrolyte films. The simulation parameters were adjusted according to the nature of the multicluster fractal patterns identified in their selected regions. In different regions, it is found that there are multiple clusters of different sizes, shapes and density. It is ascertained that for every region, there lies a governing cluster that is surrounded by other neighboring clusters. To the best of the author's knowledge, this kind of simulation work is the first and only one that has successfully been done.

In this work, fractal dimension, D for the experimental and simulated patterns for the single and multicluster fractal patterns are mostly in the range of 1.69 to 1.79. These dimension values depend on the size of the fractals. The determination of fractal dimension values was done by applying calculations based on mathematical description of dimension. Many different notions of dimension such as information dimension, mass dimension and box count dimension were discussed in this thesis. The box count dimension was chosen as it provides researchers a more efficient way to calculate the

fractal dimension. The box counting method gives a systematic measurement, which can be applied to any structure in 2D or in a 3D space.

Furthermore, the error values and percentage of difference of the fractal dimension values from comparison between the experimentally cultured fractals and the simulated fractals were found to be marginally close. It is evident that this research work has successfully come up with simulation results which are in good conformity when compared to the digital images taken from the experimental results.

7.2 Suggestions for Future Work

Although useful, the introduction of all the three simulation parameters simultaneously in the simulation of single and multicluster fractal patterns can further reach its full potential. There are still plenty of further extrusions to be explored. The simulation of DLA patterns of single and multicluster fractal patterns written in Matlab version 7.12 nevertheless has a run time restriction. To actually simulate a larger DLA pattern requires a longer time to complete as a particle that moves close to the cluster has to investigate all neighbor sites, whether these already belong to the cluster and the particle should stick or whether it can walk freely. The information about the neighborhood should be assigned to each site, so that a walker has only made contact to the site which it is on instead of for instance, all four (in 4 lattice) possible neighbors.

The study of fractal growth on films such as done in this work is useful in understanding the movement of ions in the films especially its application in the domain of batteries, nanowires and thin films. For future research, modification to the polymer-salt system

such as adding inert filler or using other type of polymer electrolytes can be considered. In addition a 3D simulation of similar fractal pattern can also be considered for further investigations. The results may be interesting and worth investigating and can contribute to the expansion of knowledge and understanding of the fractal growth phenomena.

REFERENCES

AHARONY, A. 1991. *Fractals and Disordered Systems*, Springer.

AKUEZUE, H. & STRINGER, J. 1989. Random aggregation and random-walking center of mass. *Journal of Statistical Physics*.

ALFONSECA, M. & ORTEGA, A. 2001. Determination of fractal dimensions from equivalent L systems. *IBM J. RES. & DEV,* 45**,** 797-850.

AMIR, S., ALI, S. A. H. & MOHAMED, N. S. B. 2011. Studies of fractal growth patterns in poly (ethylene oxide) and chitosan membranes. *Ionics,* 17**,** 121-125.

AMIR, S., MOHAMED, N. S. & ALI, S. A. H. 2010. Simulation model of the fractal patterns in ionic conducting polymer films. *Central European Journal of Physics,* 8**,** 150-156.

AROF, A. & MOHAMED, N. 1995. Fractals in silver nitrate added chitosan acetate films. *Jurnal Sains,* 3**,** 375-384.

BANKAR, D. N., GADE, P. M., LIMAYE, A. V. & BANPURKAR, A. G. 2007. Segregation of fractal aggregates grown from two seeds. *Physical Review E* 75**,** 051401.

BARKEY, D. 1991. Morphology Selection and the Concentration Boundary-Layer in Electrochemical Deposition. *Journal of the Electrochemical Society,* 138**,** 2912-2917.

BARNSLEY, M. F. & HAWLEY, R. 1993. *Fractals Everywhere,* Boston, Academic Press Professional.

BEN-JACOB, E., SHOCHET, O., TENENBAUM, A., COHEN, I., CZIROK, A. & VICSEK, T. 1994. *Nature,* 368**,** 46.

BHATTACHARYA, B., UPADHYAYA, H. & CHANDRA, S. 1996. Photoelectrochemical studies of an ion conducting polymer (PEO)/semiconductor (Si) junction. *Solid State Communications,* 98**,** 633-638.

BIEHL, M. 2005. *Lattice gas models and Kinetic Monte Carlo simulations of epitaxial growth In:Multiscale Modeling in Epitaxial Growth*, Birkhaeuser.

BROADBENT, S. R. & HAMMERSLEY, J. M. 1957. Percolation Process I. Crystals and Mazes. . *Proc. Cambridge Philos. Soc. ,* 53**,** 629-641.

BRUNNER, R., GALL, S., WILKE, W. & ZRINYI, M. 1995. FORMATION OF FRACTAL STRUCTURES BY AGGREGATION OF ANISOMETRIC IRON(III) HYDROXIDE PARTICLES. *Physica a-Statistical Mechanics and Its Applications,* 214**,** 153-161.

BUNDE, A. & HAVLIN, S. 1991. *Fractals and Disordered Systems*, Springer, Berlin.

CARR, B. J. & COLEY, A. A. 2003. Self-similarity in general relativity. Available: http://users.math.uni-potsdam.de/~oeitner/QUELLEN/ZUMCHAOS/selfsim1.htm [Accessed 2/3/2010].

CATALAN, G., BEA, H., FUSIL, S., BIBES, M., PARUCH, P., BARTHELEMY, A. & SCOTT, J. F. 2008. Fractal dimension and size scaling of domains in thin films of multiferroic BiFeO3. *Physical Review Letters,* 100.

CHANDRA, A. 1996. Anion clustering and fractal pattern growth in ion conducting polymeric matrix. *Solid State Ionics,* 86-8**,** 1437-1442.

CHANDRA, A. & CHANDRA, S. 1994. Experimental-Observation of Large-Size Fractals in Ion-Conducting Polymer Electrolyte Films. *Physical Review B,* 49**,** 633-636.

CHATTARAJ, P. P., KALIDAHA, A. K., MUKHOPADHYAY, R., BHATACHARYA, A. K. & TRIPATHY, D. K. 1996. Rheological study of filled SBR compounds with trans-polyoctenylene (TOR) and their interaction mechanism. *International Journal of Polymeric Materials,* 33**,** 73-87.

CHEN, J. D. & WILKINSON, D. 1985. Pore-Scale Viscous Fingering in Porous-Media. *Physical Review Letters,* 55**,** 1892-1895.

COMBES, F. 1998. Fractal Structures Driven by Self-gravity: Molecular Clouds and the Universe. *Celestial Mechanics and Dynamical Astronomy,* 72**,** 1-2.
CROSS, S. S. 1997. Fractals in pathology. *Journal of Pathology,* 182**,** 1-8.
CURTIS, C. W. 1999. *Pioneers of representation theory: Frobenius, Burnside, Schur, and Brauer*, American Mathematical Society and London Mathematical Society.
DARGAHI-NOUBARY, G. R. 1997. A test of the cyclicity of earthquakes. *Natural Hazards,* 16**,** 127-134.
DEERING, W. & WEST, B. J. 1992. Fractal Physiology. *IEEE Engin. Med. Biol.,* 11**,** 40-46.
DIERKING, I. 2001. Fractal growth of the liquid crystalline B2 phase of a bent-core mesogen. *Journal of Physics-Condensed Matter,* 13**,** 1353-1360.
DURER, A. 1525. The Painter's Manual. Available: http://en.wikipedia.org/wiki/Albrecht_Durer [Accessed 1/1/2010].
EBERT, D. S. 1996. Advanced Modeling Techniques for Computer Graphics. *ACM Computing Surveys,* 28**,** 153-1556.
EDEN, M. 1961. *Proc. 4th Berkeley Symp. on Math. Stat. and Prob,* 4**,** 223.
FALCONER, K. 2003. *Fractal Geometry: Mathematical Foundations and Applications*, John Wiley & Sons.
FAMILY, F. 1990. Dynamic Scaling and Phase-Transitions in Interface Growth. *Physica A,* 168**,** 561-580.
FAMILY, F. & VICSEK, T. 1985. Scaling of the Active Zone in the Eden Process on Percolation Networks and the Ballistic Deposition Model. *Journal of Physics a-Mathematical and General,* 18**,** L75-L81.
FEDER, J. 1988. *Fractals, Physics of solids and liquids,* New York, Plenum Press.
FOROUTAN-POUR, K., DUTILLEUL, P. & SMITH, D. L. 1999. Advances in the implementation of the box-counting method of fractal dimension estimation. *Applied Mathematics and Computation,* 105**,** 195-210.
GOLUBEV, Y., FOMIN, V. & CHERKESOV, L. 1987. Interaction between surface gravitational waves and local rise of sea-bed in the uniform ocean. *Physical Oceanography,* 1.
GREGORY, J. 1998. The role of floc density in solid-liquid separation. *Filtration & Separation,* 35**,** 367-371.
GULYÁS, A. & SZEDENIK, N. 2009. 3D simulation of the lightning path using a mixed physical-probabilistic model – The open source lightning model *Journal of Electrostatics,* 67**,** 518-523.
HAC, A. E., SEEGER, H. M., FIDORRA, M. & HEIMBURG, T. 2005. Diffusion in two-component lipid membranes - A fluorescence correlation spectroscopy and Monte Carlo simulation study. *Biophysical Journal,* 88**,** 317-333.
HASHIM ALI, S. A., MOHAMED, N. S., SHARIFF, A. A. & AROF, A. K. 2000. *Solid State Ionic Devices: Science & Technology***,** 16-19.
HASTINGS, H. & SUGIHARA, G. 1993. *Fractals a user's guide for the natural sciences*, University Press.
HE, L. & HUANG, L. 2008. Pattern formation on a stretchable substrate. *International Journal of Solids and Structures.*
HENTSCHEL, H. 1992. Interfaces driven by quenched random fields. *Physical Review A,* 46.
HORVATH, V. K., FAMILY, F. & VICSEK, T. 1991. Anomalous Noise Distribution of the Interface in 2-Phase Fluid-Flow. *Physical Review Letters,* 67**,** 3207-3210.
IRURZUN, I., BERGERO, P., MOLA, V., CORDERO, M., VICENTE, J. & MOLA, E. 2002. Dielectric breakdown in solids modeled by DBM and DLA. *Chaos, Solitons & Fractals.*
JAIN, A. 1989. *Fundamentals of digital image processing,* New Jersey, Prentice Hall.
JANKE, W. & SCHAKEL, A. 2005. Fractal structure of high-temperature graphs of O (n) models in two dimensions. *Phys Rev Lett,* 95**,** 135702.

JELINEK, H. F. & FERNANDEZ, E. 1998. Neurons and fractals: how reliable and useful are calculations of fractal dimensions? *Journal of Neuroscience Methods,* 81**,** 9-18.
JUDD, C. 2003. Fractals – Self-Similarity. Available: http://www.bath.ac.uk/~ma0cmj/FractalContents.html [Accessed 31/12/2009].
KAANDORP, J. 1994. *Fractal modelling: growth and form in biology*, Springer-Verlag.
KAHANE, J. P. 1983. *Ensembles aleatoires et dimensions: (Cours au seminaire de l'Escurial, 30 juin-5 juillet 1983)*, Departement de mathematique.
KAUFMAN, J. H., NAZZAL, A. I., MELROY, O. R. & KAPITULNIK, A. 1987. Onset of Fractal Growth - Statics and Dynamics of Diffusion-Controlled Polymerization. *Physical Review B,* 35**,** 1881-1890.
KENKEL, N. C. & WALKER., D. J. 1993. Fractals and ecology. *Abst. Bot. ,* 17**,** 53-70.
KNUDSEN, H., SANDNES, B., FLEKKØY, E. & MÅLØY, K. 2008. Granular labyrinth structures in confined geometries. *Physical Review E,* 77.
LEHTINEN, K. E. J., WINDELER, R. S. & FRIEDLANDER, S. K. 1996. Prediction of nanoparticle size and the onset of dendrite formation using the method of characteristic times *Journal of Aerosol Science,* 27**,** 883-896.
LI, X. L., YAO, K. L. & LIU, Z. L. 2009. Cluster Moving Monte Carlo Simulation of Nano-Sized Magnetic Particle Aggregation in an Applied Magnetic Field. *International Journal of Modern Physics B,* 23**,** 5307-5323.
LIANG, S. 1986. Random-Walk Simulations of Flow in Hele Shaw Cells. *Physical Review A,* 33**,** 2663-2674.
LITWINENKO, J. W., ROJAS, A. M., GERSCHENSON, L. N. & MARANGONI, A. G. 2002. Relationship between crystallization behavior, microstructure, and mechanical properties in a palm oil-based shortening. *Journal of the American Oil Chemists Society,* 79**,** 647-654.
LO VERSO, F., VINK, R. L. C., PINI, D. & REATTO, L. 2006. Critical behavior in colloid-polymer mixtures: Theory and simulation. *Physical Review E,* 73.
LOEHLE, C. 1983. The fractal dimension and ecology *Specul. Sci. Tech.,* 6**,** 131-142.
MALOY, K. J., FEDER, J. & JOSSANG, T. 1985. *Phys. Rev. Lett.,* 55**,** 2688.
MANDELBROT, B. 1977. *Fractals: Form, Chance and Dimension*, W H Freeman and Co.
MANDELBROT, B. 1983. *The fractal geometry of nature,* New York, Wh Freeman.
MARCH, P. 1992. Remarks on scaling a model of Witten-Sander type. *Journal of Statistical Physics*.
MARCONE, B., ORLANDINI, E. & STELLA, A. L. 2007. Knot localization in adsorbing polymer rings. *Physical Review E,* 76.
MATSUSHITA, M., F.HIRAMATSU, N.KOBAYASHI, OZAWA, T., YAMAZAKI, Y. & MATSUYAMA, T. 2004. Colony formation in bacteria: experiments and modeling. *Biofilms,* 1**,** 305-317.
MATSUSHITA, M., SANO, M., HAYAKAWA, Y., HONJO, H. & SAWADA, Y. 1984a. *Phys. Rev. Lett.,* 53**,** 286.
MATSUSHITA, M., SANO, M., HAYAKAWA, Y., HONJO, H. & SAWADA, Y. 1984b. Fractal structures of zinc metal leaves grown by electrodeposition. *Phys Rev Lett*.
MATSUURA, S. & MIYAZIMA, S. 1992. *Physica A,* 191**,** 30.
MATSUYAMA, T., HARSHEY, R. M. & MATSHUSHITA, M. 1993. *Fractals,* 1**,** 336.
MCNAMEE, J. E. 1991. Fractal Perspectives in Pulmonary Physiology. *J. Appl. Physiol.,* 71**,** 1-8.
MEAKIN, P. 1983. Formation of fractal clusters and networks by irreversible diffusion-limited aggregation. *Phys Rev Lett*.
MEAKIN, P. 1988. Reaction-Limited Cluster-Cluster Aggregation in Dimensionalities 2-10. *Physical Review A,* 38**,** 4799-4814.
MEAKIN, P. 1991. Models for Material Failure and Deformation. *Science,* 252**,** 226–234.
MEAKIN, P., FAMILY, F. & VICSEK, T. 1987. Viscous Fingering Simulated by Off-Lattice Aggregation. *J Colloid Interface Sci,* 117**,** 394-399.

MILOSEVIC, N. T. & RISTANOVIC, D. 2007. Fractal and nonfractal properties of triadic Koch curve. *Chaos Solitons & Fractals,* 34**,** 1050-1059.
MOHAMED, N. & AROF, A. 2001. Fractal like dendritic crystals of lithium tetrafluoroborate in chitosan acetate films. *Malaysian Journal of Analytical Sciences,* 6**,** 71-74.
MUKHERJEE, S., JACOBS, D. & NAKANISHI, H. 1995. Diffusion on loopless critical percolation cluster. *Journal of Physics A:Math. Gen,* 28**,** 291-296.
MUSGRAVE, F. K. 1993. *Methods for Realistic Landscape Imaging*, Yale University.
NIEMEYER, L., PIETRONERO, L. & WIESMANN, H. J. 1984. *Phys. Rev. Lett. ,* 52**,** 1033.
NIEMEYER, L., PIETRONERO, L., WIESMANN, H.J. 1984. Fractal dimension of dielectric breakdown *Physical Review Letters,* 52**,** 1033-1036.
OKUBO, S., MOGI, I., KIDO, G. & NAKAGAWA, Y. 1993. Effect of High Magnetic Fields on Fractal Growth of Silver Metal-Forest. *Fractals-Complex Geometry Patterns and Scaling in Nature and Society,* 1**,** 425-429.
OSSADNIK, P., LAM, C. H. & SANDER, L. M. 1994. Nonuniversal Diffusion-Limited Aggregation and Exact Fractal Dimensions. *Physical Review E,* 49**,** R1788-R1791.
OUELLETTE, J. 2001. Pollock'sFractals. Available: http://discovermagazine.com/2001/nov/featpollock [Accessed 3/3/2010].
PATERSON, L. 1984. Diffusion-limited aggregation and two-fluid displacements in porous media. *Phys Rev Lett.*
PEITGEN, H.-O., JÜRGENS, H. & SAUPE, D. 2004. *Chaos and Fractals: New Frontiers of Science* New York, Springer Verlag.
PEITGEN, H. O. & SAUPE, D. 1988. *The Science of Fractal Images,* Berlin, Springer-Verlag.
PRAUD, O. & SWINNEY, H. 2005. Fractal dimension and unscreened angles measured for radial viscous fingering. *Physical Review E,* 72**,** 011406.
RADNOCZY, G., VICSEK, T, SANDER, L.M. & GRIER, D. 1987. *Phys.Rev A,* 35**,** 4012.
RATHGEBER, S., MONKENBUSCH, M., HEDRICK, J. L., TROLLSAS, M. & GAST, A. P. 2006. Starlike dendrimers in solutions: Structural properties and internal dynamics. *Journal of Chemical Physics,* 125.
RIBEIRO, L. M. F., HOROVISTIZ, A. L., JESUINO, G. A., HEIN, L. R. D., ABBADE, N. P. & CRNKOVIC, S. J. 2002. Fractal analysis of eroded surfaces by digital image processing. *Materials Letters,* 56**,** 512-517.
ROSSO, M. 2007. Electrodeposition from a binary electrolyte: new developments and applications. *Electrochimica Acta,* 53**,** 250-256.
RUCKER, R. 1984. *The Fourth Dimension*, Houghton-Mifflin.
SAFFMAN, P. G. & TAYLOR, G. 1958. The penetration of a fluid into a medium of hele-shaw cell containing a more viscous liquid. *Proc. Soc. London,* Ser A**,** 312-329.
SAWADA, Y., DOUGHERTY, A. & GOLLUB, J. P. 1986. *Phys. Rev. Lett. ,* 56**,** 1260.
SHAIKH, Y. H., KHAN, A. R., PATHAN, J. M., PATIL, A. & BEHERE, S. H. 2009. Fractal pattern growth simulation in electrodeposition and study of the shifting of center of mass. *Chaos, Solitons and Fractals,* 42**,** 2796-2803.
SHIBKOV, A., GOLOVIN, Y., ZHELTOV, M., KOROLEV, A. & VLASOV, A. 2001. Kinetics and morphology of nonequilibrium growth of ice in supercooled water. *Crystallography Reports,* 46**,** 496-502.
SHUI, J., JIANG, G., XIE, S. & CHEN, C. 2004. Thin films of lithium manganese oxide spinel as cathode materials for secondary lithium batteries. *Electrochimica Acta,* 49.
SŁAWIŃSKI, C., SOKOŁOWSKA, Z., WALCZAK, R. & ... 2002. Fractal dimension of peat soils from adsorption and from water retention experiments. *Colloids and Surfaces A:Physicochemical and Engineering Aspects,* 208**,** 289-301.
SMITH, J. T. G., MARKS, W. B., LANGE, G. D., JR., W. H. S. & NEALE, E. A. 1990. A fractal analysis of cell images. *J.Neurosci.Meth.,* 27**,** 173-180.

SOMASUNDARAN, P. & RUNKANA, V. 2003. Modeling flocculation of colloidal mineral suspensions using population balances. *International Journal of Mineral Processing,* 72**,** 33-55.

SORENSEN, C. M. 2011. The Mobility of Fractal Aggregates: A Review. *Aerosol Science and Technology,* 45**,** 765-779.

STANLEY, H., BULDYREV, S., GOLDBERGER, A., HAVLIN, S., MANTEGNA, R., OSSADNIK, S., PENG, C., SCIORTINO, F. & SIMONS, M. 1994. Fractals in biology and medicine. *Diffusion Processes: Lecture notes in Physics.*

SUKI, M., MOHAMED, N., ALI, S. H. & ZAINUDDIN, R. 2007. *Malaysian Journal of Science***,** 23-33.

TAN, Z., ZOU, X., ZHANG, W. & JIN, Z. 1999. Influences of the size and dielectric properties of particles on electrorheological response. *Physical Review E.*

TAN, Z., ZOU, X., ZHANG, W. & JIN, Z. 2000. Structure transition in cluster-cluster aggregation under external fields. *Physical Review E.*

TANNER, J. E. 1995. Competition between scleractinian corals and macroalgae: An experimental investigation of coralgrowth, survival and reproduction. *Journal of Experimental Marine Biology and Ecology,* 190**,** 151-168.

TORDOFF, G., BODDY, L. & JONES, T. H. 2008. Species-specific impacts of collembola grazing on fungal foraging ecology. *Soil Biology and Biochemistry***,** 1-9.

VICSEK, T. 1989. *Fractal Growth Phenomena,* Riveredge, NJ USA, World Scientific.

VICSEK, T. 1992. *Fractal growth phenomena,* Singapore, World Scientific.

VILLANI, V. & COMENGES, J. M. Z. 2000. Analysis of biomolecular chaos in aqueous solution. *Theoretical Chemistry Accounts,* 104**,** 290-295.

WEST, B. J. & GOLDBERGER, A. L. 1987. Physiology in fractal dimensions. *Am. Sci. ,* 75**,** 354-365.

WIENS, J. A. 1989. Spatial scaling in ecology. *Funct. Ecol. ,* 3**,** 385-397.

WITTEN, T. & SANDER, L. 1981. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys Rev Lett.*

WITTEN, T. & SANDER, L. 1983. Diffusion-limited aggregation. *Physical Review B.*

YADEGARI, S. 2001. Self-similarity. Available: http://www.crca.ucsd.edu/~syadegar/MasterThesis/node25.html [Accessed 10/1/2010].

ZHANG, G. L., JIN, L. X., MA, Z. P., ZHAI, X. M., YANG, M., ZHENG, P., WANG, W. & WEGNER, G. 2008. Dendritic-to-faceted crystal pattern transition of ultrathin poly(ethylene oxide) films. *Journal of Chemical Physics,* 129.

ZHANG, J. & LIU, Z. 1998. Study of the relationship between fractal dimension and viscosity ratio for viscous fingering with a modified DLA model. *Journal of Petroleum Science and Engineering.*

ZHI-QIANG ZOU, W.-C. L. 2011. Two-dimensional fractal-like growth on semiconductors: The formation of continuous manganese monosilicide ultrathin films on Si(111). *Physics Letters A,* 375**,** 849-854.

ZHOU, H. W., ZHANG, Y. H., LI, A. M. & QIU, D. Y. 2008. Experimental study on moving boundaries of fluid flow in porous media. *Chinese Science Bulletin,* 53**,** 2438-2445.

APPENDIX A: Main Function File for the Simulation Program of the Single Cluster Fractal Pattern

```
% Subject: DLA model with single seed on 3 different lattice site.
%
% Description: This function file is being called from myDLAdisplay.fig
which runs a DLA model with single seed on either 4, 6, or 8 lattice
site. There is only one output which is a list of particle coordinates
called    'singleCluster'.    There    are    four    inputs    needed;
radius   (r),   maximum   number   of   particles   (maxmass),   sticking
coefficient (alpha), and type of lattice site (nlattice). A small
function file called circle.m is needed as well. This function not
only returns coordinates which will be plotted in myDLAdisplay.fig,
but also calculates the dimension of the model each time a particle is
placed. The formula for dimension is:

                                    log(m)/log(rmax),

% where m is the maximum mass at particular moment and rmax is the
maximum radius correlate to it.
%

function singleCluster = myDLASingle(r,maxmass,alpha,nlattice)

% Lattice size (latt) and center site position (cp):
latt = 1001;
cp   = floor(latt/2);

% Create a region/area of a lattice site and set the center element as
occupied = 1.
coor = zeros(latt,latt);
coor(cp,cp) = 1;

% Mass (m) and maximum radius (rmax):
m = 1;
rmax = 1;

% Create columns that list coordinates (x,y) of occupied site and
dimensions respectively.
% Set initial values (first row):
dlapoints(1,:) = [cp cp log(m)/log(rmax)];


% --- LOOP TO PLACE PARTICLES ---
% Release the particles one by one
for ipart = 2: maxmass

    % Set all the conditions
    rstart = rmax + r;          % radius where the particle is released
(starting point)
    rjump  = rstart + 10;       % radius where the particle takes a huge
leap towards the center
    rkill  = 10 * rmax + 10;    % radius of boundary
    % NOTE: These radius changes as rmax increases.

    % Determine the starting point:
    pos = circle(rstart,cp,cp);     % return pos = [x y]
    ix = pos(1);
    iy = pos(2);

    % RELEASE!
```

```
    check  = 0;                % particle is free to move
    while check == 0             % check becomes nonzero at aggregation
sites

        % Distance between current position and the center (cp,cp).
        distance = sqrt((ix-cp)^2 + (iy-cp)^2);

        % CONDITION 1: distance > rkill, re-launch particle.
        if distance > rkill
           pos = circle(rstart,cp,cp);
           ix  = pos(1);
           iy  = pos(2);
           distance = sqrt((ix-cp)^2 + (iy-cp)^2);
        end %end CONDITION 1

        % CONDITION 2: distance < rjump, move particle to adjacent site.
        if distance < rjump
           check_1 = 0;      % Current position is empty.
           oldx    = ix;
           oldy    = iy;

           % Particle will ONLY move if the new position is empty.
           while check_1 == 0

               % First choose which type of lattice are used.
               if nlattice == 4
                   zz = floor(4*rand);
               else if nlattice == 6
                       zz = floor(6*rand);
                   else if nlattice == 8
                           zz = floor(8*rand);
                       end
                   end
               end %end if nlattice

               switch(zz)
                   case 0,
                       ix = ix+1;
                   case 1,
                       ix = ix-1;
                   case 2,
                       iy = iy+1;
                   case 3,
                       iy = iy-1;
                   case 4,
                       ix = ix+1;
                       iy = iy+1;
                   case 5,
                       ix = ix-1;
                       iy = iy-1;
                   case 6,
                       ix = ix-1;
                       iy = iy+1;
                   case 7,
                       ix = ix+1;
                       iy = iy-1;
               end  %end switch

               if coor(ix,iy) == 0
                  check_1 = 1;
               else
                   ix = oldx;
                   iy = oldy;
```

```
            end  %end if

        end  %end while check_1

    % CONDITION 3:  rjump < distance < rkill, leap further from
current position.
    else
        pos = circle((distance-rstart),ix,iy);

        % Jump to any direction with given radius from current
position.
        ix = pos(1);
        iy = pos(2);
    end  %end CONDITION 2 & 3

    distance = sqrt((ix-cp)^2 + (iy-cp)^2);

% NOTE: SET CHECK = 1 IF A NN SITE OF CLUSTER IS REACHED!

    % CONDITION 4: distance < rstart, check if the coordinate is not
occupied.
    if distance < rstart

        % Again, choose which type of lattice are used.
        if nlattice == 4
            check_2 = coor(ix-1,iy) + coor(ix+1,iy) + ...
                      coor(ix,iy-1) + coor(ix,iy+1);
        else if nlattice == 6
                check_2 = coor(ix-1,iy) + coor(ix+1,iy) + ...
                          coor(ix,iy-1) + coor(ix,iy+1) + ...
                          coor(ix+1,iy+1) + coor(ix-1,iy-1);
            else if nlattice == 8
                    check_2 = coor(ix-1,iy) + coor(ix+1,iy) + ...
                              coor(ix,iy-1) + coor(ix,iy+1) + ...
                              coor(ix-1,iy+1) + coor(ix+1,iy+1) + ...
                              coor(ix-1,iy-1) + coor(ix+1,iy-1);
                end
            end
        end %end if nlattice

        if check_2 ~= 0
            if alpha == 0
                check = 1;
            else
                arandom = rand;
                if arandom <= alpha
                    check = 1;
                else
                    check = 0;
                end
            end
        end

    end  %end CONDITION 4

end % end of while check == 0

% Set the coordinate as occupied site = 1.
coor(ix,iy) = 1;

% Once the particle stick, determine mass and maximum radius:
m = m+1;
rmax = max(rmax,sqrt((ix-cp)^2 + (iy-cp)^2));
```

```
    % Update the values needed into a list.
    dlapoints(ipart,:) = [ix iy log(m)/log(rmax)];

end  %end for ipart

singleCluster = dlapoints;

% Callback from myDLAdisplay.fig
setappdata(0, 'singleCluster', singleCluster);
setappdata(0, 'coor', coor);  % Needed for boxcount

end
```

```
Input argument "maxmass" is undefined.

Error in ==> myDLASingle at 47
for ipart = 2: maxmass
```

APPENDIX B: Graphical User Interface (GUI) Program File for the Simulation Program of the Single Cluster Fractal Pattern

```
% Subject: GUI for DLA model with single seed.
%
% Description:  A GUI program that displays DLA model with single seed.
%               There are 3 inputs needed (2 edit text, 1 pop-up menu),
%                 2 outputs which display on axes, as well as 3 buttons
and
%               1 checkbox to control the output. User must type all the
%               inputs needed and click on 'Start' button to run the
%                       program.  When  the  program  runs,  it  will  call
myDLASingle.m
%               function file and plot the coordinates obtained. The
%               outputs display in the form of fractal image on the main
axis and
%                   a  linear  graph  on  log(m)  vs.  log(rm)  to  obtain  the
avarage dimension.
%               There is also a 'BoxCount' button to attain dimension of
%                  fractal using boxcount method when it is clicked. A
counter
%               is also included to show the increment of mass (no. of
%               particles), maximum radius and dimension as one by one
%               particle is placed.
%-------------------------------------------------------------------------
------------

function varargout = myDLAdisplay(varargin)
% MYDLADISPLAY M-file for myDLAdisplay.fig
%      MYDLADISPLAY, by itself, creates a new MYDLADISPLAY or raises the
existing
%      singleton*.
%
%      H = MYDLADISPLAY returns the handle to a new MYDLADISPLAY or the
handle to
%      the existing singleton*.
%
%      MYDLADISPLAY('CALLBACK',hObject,eventData,handles,...) calls the
local
%       function named CALLBACK in MYDLADISPLAY.M with the given input
arguments.
%
%       MYDLADISPLAY('Property','Value',...) creates a new MYDLADISPLAY
or raises the
%        existing  singleton*.   Starting  from  the  left,  property  value
pairs are
%       applied to the GUI before myDLAdisplay_OpeningFcn gets called.
An
%         unrecognized  property  name  or  invalid  value  makes  property
application
%        stop.   All  inputs  are  passed  to  myDLAdisplay_OpeningFcn  via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help myDLAdisplay
```

```
% Last Modified by GUIDE v2.5 05-Mar-2012 16:43:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @myDLAdisplay_OpeningFcn, ...
                   'gui_OutputFcn',  @myDLAdisplay_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin(Vicsek))
    gui_State.gui_Callback = str2func(varargin(Biehl));
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before myDLAdisplay is made visible.
function myDLAdisplay_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no main_output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to myDLAdisplay (see VARARGIN)

set(handles.figure1, 'MenuBar', 'figure','Name','My DLA Model');

% Choose default command line main_output for myDLAdisplay
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(handles.start_pushbutton,'UserData',0);

% UIWAIT makes myDLAdisplay wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = myDLAdisplay_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning main_output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line main_output from handles structure
varargout{1} = handles.output;


% ------------------------------ INPUTS ----------------------------------
---
function input_maxparticles_Callback(hObject, eventdata, handles)
% hObject    handle to input_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input_alpha as text
%               str2double(get(hObject,'String'))  returns  contents  of
input_alpha as a double
```

```
%store the contents of input_alpha as a string. if the string
%is not a number then input will be empty
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input_alpha to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function input_maxparticles_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function input_alpha_Callback(hObject, eventdata, handles)
% hObject    handle to input_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of input_alpha as text
%               str2double(get(hObject,'String'))  returns  contents  of
input_alpha as a double

%store the contents of input_alpha as a string. if the string
%is not a number then input will be empty
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input_alpha to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function input_alpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function lattice_popupmenu_Callback(hObject, eventdata, handles)

function lattice_popupmenu_CreateFcn(hObject, eventdata, handles)
```

```
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --------------------------- END INPUTS ----------------------------
---


% ------------------------------ MAIN --------------------------------
---
% --- Executes on button press in start_pushbutton.
function start_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to start_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

tic
if get(handles.start_pushbutton,'UserData')==1, return; end
set(handles.start_pushbutton,'UserData',1);

set(handles.start_pushbutton, 'Enable', 'Off');
set(handles.pause_checkbox, 'Enable', 'On');
set(handles.stop_pushbutton,  'Enable', 'On');
set(handles.clear_pushbutton, 'Enable', 'Off');

maxparticlesValue                                                    =
str2double(get(handles.input_maxparticles,'string'));
alphaValue        = str2double(get(handles.input_alpha,'string'));

% If user enters a wrong value, a warning box will appear.
% The value of sticking coefficient (alpha) must be in the range of
[0,1]
if (alphaValue > 1) || (alphaValue < 0)
    w = warndlg(sprintf('Sticking coefficient must be in range [0,1] \n
Please re-enter the value'), 'Warning');
    waitfor(w);
    set(handles.input_alpha,'String','0');

    set(handles.start_pushbutton, 'Enable', 'On');
    set(handles.pause_checkbox, 'Enable', 'Off');
    set(handles.stop_pushbutton, 'Enable', 'Off');
    set(handles.clear_pushbutton, 'Enable', 'On');

    set(handles.start_pushbutton,'UserData',0);
    return
end

% Obtained value from pop-up menu
list      = get(handles.lattice_popupmenu,'String');
val       = get(handles.lattice_popupmenu,'Value');
noLattice = str2double(list{val});

% Call a function to obtain coordinates
myDLASingle(5,maxparticlesValue,alphaValue,noLattice);
singleCluster = getappdata(0, 'singleCluster');
toc

tic
% Plot the coordinates
% Select main_output as the current axes, so that Matlab knows where to
plot
```

```
axes(handles.main_output)

m = 1;
rmax = 1;

for j = 1:maxparticlesValue

    % Each time a user click on 'Stop' button, the program terminates.
    if get(handles.start_pushbutton,'UserData')==0, break;
    end

    if j == 1

        % Plot center point with red dot.
        plot(singleCluster(j,1),singleCluster(j,2),'r*');
        hold on

    else

        % Plot the rest of the points with green dot.
        x = singleCluster(j,1);
        y = singleCluster(j,2);

        plot(x,y,'g.');
        hold on

        m = m+1;
        rmax = max(rmax,sqrt((singleCluster(j,1)-singleCluster(1,1))^2 +
...
                  (singleCluster(j,2)-singleCluster(1,2))^2));

    end  %end if

    set(gca,'ytick',[]);
    set(gca,'xtick',[]);

    % Calculate the dimension
    dim = log(m)/log(rmax);

    % Update counter
    set(handles.massCounter, 'String', num2str(m));
    set(handles.radiusCounter, 'String', num2str(rmax));
    set(handles.dimCounter, 'String', num2str(dim));

    pause(.0001);
end  %end for

% Plot the dimension graph
% Select graph_log as the current axes, so that Matlab knows where to
plot
axes(handles.graph_log)

m1  = zeros(maxparticlesValue,1);
rm1 = zeros(maxparticlesValue,1);

m1(1)  = 1;
rm1(1) = 1;

for i = 2:maxparticlesValue

    m1(i) = m1(i-1) + 1;
    rm1(i) = max(rm1(i-1),sqrt((singleCluster(i,1)-singleCluster(1,1))^2
+ ...
```

```
            (singleCluster(i,2)-singleCluster(1,2))^2));
end

p = log(rm1);
q = log(m1);

plot(p, q,'*');
hold on
F = fit(p,q,'poly1')
plot(F);

legend off;
xlabel('log(rm)');
ylabel('log(m)');

P = polyfit(p,q,1);
set(handles.avrDim, 'String', num2str(P(1)));

set(handles.pause_checkbox, 'Enable', 'Off');
set(handles.stop_pushbutton, 'Enable', 'Off');
set(handles.clear_pushbutton, 'Enable', 'On');
toc

guidata(hObject, handles); %updates the handles
set(handles.start_pushbutton,'UserData',0);

% ------------------------------ END MAIN --------------------------------
---


% ------------------------------ BUTTONS ---------------------------------
---
% --- Executes on button press in clear_pushbutton.
function clear_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to clear_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.start_pushbutton,  'Enable', 'On');

% Clears everything : both input and both axes
set(handles.input_maxparticles,'string','0');
set(handles.input_alpha,'string','0');
set(handles.lattice_popupmenu,'value',1)
set(handles.massCounter, 'string', '0');
set(handles.radiusCounter, 'string', '0');
set(handles.dimCounter, 'string', '0');
set(handles.avrDim, 'string', '0');

cla(handles.main_output);
cla(handles.graph_log);

guidata(hObject, handles); %updates the handles


% --- Executes on button press in pause_checkbox.
function pause_checkbox_Callback(hObject, eventdata, handles)
% hObject    handle to pause_checkbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of pause_checkbox
```

```
%checkboxStatus = 0, if the box is unchecked,
%checkboxStatus = 1, if the box is checked
checkboxStatus = get(handles.pause_checkbox,'Value');

if(checkboxStatus)
    set(handles.start_pushbutton, 'Enable', 'Off');
    set(handles.stop_pushbutton,  'Enable', 'Off');
    set(handles.clear_pushbutton, 'Enable', 'Off');

    %if box is checked, pause loop
    uiwait(ancestor(handles.main_output,'figure'));

    guidata(hObject, handles);

else
    set(handles.clear_pushbutton,  'Enable', 'Off');
    set(handles.stop_pushbutton, 'Enable', 'On');

    %if box is unchecked, continue loop
    uiresume(ancestor(handles.main_output,'figure'));


    guidata(hObject, handles);
end


% --- Executes on button press in stop_pushbutton.
function stop_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to stop_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.start_pushbutton,'UserData',0);

set(handles.pause_checkbox, 'Enable', 'Off');
set(handles.stop_pushbutton, 'Enable', 'Off');
set(handles.clear_pushbutton, 'Enable', 'On');

guidata(hObject, handles);


function boxcount_pushbutton_Callback(hObject, eventdata, handles)

c = getappdata(0, 'coor');
boxcount(c,'slope')

guidata(hObject, handles);
% ---------------------------- END BUTTONS ----------------------------
---


% ---------------------------- COUNTER --------------------------------
---
function massCounter_Callback(hObject, eventdata, handles)
% hObject    handle to massCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of massCounter as text
%               str2double(get(hObject,'String'))  returns  contents  of
massCounter as a double

% --- Executes during object creation, after setting all properties.
function massCounter_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to massCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function radiusCounter_Callback(hObject, eventdata, handles)
% hObject    handle to radiusCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of radiusCounter as text
%               str2double(get(hObject,'String')) returns contents of
radiusCounter as a double

% --- Executes during object creation, after setting all properties.
function radiusCounter_CreateFcn(hObject, eventdata, handles)
% hObject    handle to radiusCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function dimCounter_Callback(hObject, eventdata, handles)
% hObject    handle to dimCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dimCounter as text
%               str2double(get(hObject,'String')) returns contents of
dimCounter as a double

% --- Executes during object creation, after setting all properties.
function dimCounter_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dimCounter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function avrDim_Callback(hObject, eventdata, handles)
```

```
function avrDim_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% ------------------------------ END COUNTER --------------------------
---
```

APPENDIX C: Programming code of the Multiple Cluster Fractal Pattern

```
% Subject: GUI for DLA model with multiple seeds.
%
% Description:   A GUI program that displays DLA model with multiple
seeds.
%               There are 4 inputs needed (3 edit text, 1 pop-up menu),
%                 1 output which displays on axis, as well as 3 buttons
and
%               1 checkbox to control the output. User must type all the
%               inputs needed and click on 'Start' button to run the
%                program. When the program runs, it will call a function
from
%                myDLAMultiple.m file and plot the coordinates obtained.
The
%               output displays in the form of fractal image on an axis.
%-------------------------------------------------------------------------
------------


function varargout = multiDLAmodified(varargin)
% MULTIDLAMODIFIED MATLAB code for multiDLAmodified.fig
%       MULTIDLAMODIFIED, by itself, creates a new MULTIDLAMODIFIED or
raises the existing
%      singleton*.
%
%      H = MULTIDLAMODIFIED returns the handle to a new MULTIDLAMODIFIED
or the handle to
%      the existing singleton*.
%
%       MULTIDLAMODIFIED('CALLBACK',hObject,eventData,handles,...) calls
the local
%        function named CALLBACK in MULTIDLAMODIFIED.M with the given
input arguments.
%
%             MULTIDLAMODIFIED('Property','Value',...)  creates  a  new
MULTIDLAMODIFIED or raises the
%        existing singleton*.  Starting from the left, property value
pairs are
%         applied to the GUI before multiDLAmodified_OpeningFcn gets
called.  An
%         unrecognized property name or invalid value makes property
application
%       stop.  All inputs are passed to multiDLAmodified_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help multiDLAmodified

% Last Modified by GUIDE v2.5 06-Mar-2012 13:50:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @multiDLAmodified_OpeningFcn, ...
```

```
                   'gui_OutputFcn',  @multiDLAmodified_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before multiDLAmodified is made visible.
function   multiDLAmodified_OpeningFcn(hObject,   eventdata,   handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to multiDLAmodified (see VARARGIN)

set(handles.figure1,  'MenuBar',  'figure','Name','DLA  Model  (Multiple
Seeds)');

% Choose default command line output for multiDLAmodified
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(handles.start_pushbutton,'UserData',0);

% UIWAIT makes multiDLAmodified wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function  varargout  =  multiDLAmodified_OutputFcn(hObject,  eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


function start_pushbutton_Callback(hObject, eventdata, handles)

if get(handles.start_pushbutton,'UserData')==1, return; end
set(handles.start_pushbutton,'UserData',1);

set(handles.pause_checkbox, 'Enable', 'On');
set(handles.stop_pushbutton,  'Enable', 'On');
set(handles.clear_pushbutton, 'Enable', 'Off');

maxparticlesValue                                                    =
str2double(get(handles.maxParticles_input,'string'));
alphaValue        = str2double(get(handles.alpha_input,'string'));

% If user enters a wrong value, a warning box will appear.
```

```
% The value of sticking coefficient (alpha) must be in the range of
[0,1]
if (alphaValue > 1) || (alphaValue < 0)
    w = warndlg(sprintf('Sticking coefficient must be in range [0,1] \n
Please re-enter the value'), 'Warning');
    waitfor(w);
    set(handles.alpha_input,'String','0');

    set(handles.pause_checkbox, 'Enable', 'Off');
    set(handles.stop_pushbutton, 'Enable', 'Off');
    set(handles.clear_pushbutton, 'Enable', 'On');

    set(handles.start_pushbutton,'UserData',0);
    return
end


nMax = str2double(get(handles.seeds_input,'string'));

% If user enters a wrong value, a warning box will appear.
% The number of seeds (nMax) must be in the range of [1,100]
if (nMax > 100) || (nMax == 0)
    w = warndlg(sprintf('Number of seeds must be in range [1,100] \n
Please re-enter number of seeds'), 'Warning');
    waitfor(w);
    set(handles.seeds_input,'String','0');

    set(handles.pause_checkbox, 'Enable', 'Off');
    set(handles.stop_pushbutton, 'Enable', 'Off');
    set(handles.clear_pushbutton, 'Enable', 'On');

    set(handles.start_pushbutton,'UserData',0);
    return
end

% Obtained value from pop-up menu
list      = get(handles.lattice_popmenu,'String');
val       = get(handles.lattice_popmenu,'Value');
noLattice = str2double(list{val});

% Call a function to obtain coordinates
myDLAMultiple(maxparticlesValue, alphaValue, nMax, noLattice);
multiCluster = getappdata(0, 'multiCluster');
cx           = getappdata(0, 'xseed');
cy           = getappdata(0, 'yseed');

% Plot the coordinates
% Select axes1 as the current axes, so that Matlab knows where to plot
axes(handles.axes1)

% Start plotting particles one by one
for j = 1:maxparticlesValue

    % Each time a user click on 'Stop' button, the program terminates.
    if get(handles.start_pushbutton,'UserData')==0, break;
    end

    xplot = multiCluster(j,1);
    yplot = multiCluster(j,2);

    % Different colour for different cluster
    plot(xplot,yplot,'g.');
    hold on
```

```
    set(gca,'ytick',[]);
    set(gca,'xtick',[]);

    pause(0.00001);

end  %end for j

for n = 1:nMax
    figure(1);
    plot(cx(n),cy(n),'r.','MarkerSize',7);
    hold on
end

set(handles.pause_checkbox, 'Enable', 'Off');
set(handles.stop_pushbutton, 'Enable', 'Off');
set(handles.clear_pushbutton, 'Enable', 'On');

guidata(hObject, handles); %updates the handles
set(handles.start_pushbutton,'UserData',0);


% ------------------------------ BUTTONS --------------------------------
---
function pause_checkbox_Callback(hObject, eventdata, handles)
%checkboxStatus = 0, if the box is unchecked,
%checkboxStatus = 1, if the box is checked
checkboxStatus = get(handles.pause_checkbox,'Value');

if(checkboxStatus)
    set(handles.start_pushbutton, 'Enable', 'Off');
    set(handles.stop_pushbutton,  'Enable', 'Off');
    set(handles.clear_pushbutton, 'Enable', 'Off');

    %if box is checked, pause loop
    uiwait(ancestor(handles.axes1,'figure'));

    guidata(hObject, handles);

else
    set(handles.clear_pushbutton,  'Enable', 'Off');
    set(handles.stop_pushbutton, 'Enable', 'On');

    %if box is unchecked, continue loop
    uiresume(ancestor(handles.axes1,'figure'));

    guidata(hObject, handles);
end

function stop_pushbutton_Callback(hObject, eventdata, handles)

set(handles.start_pushbutton,'UserData',0);

set(handles.pause_checkbox, 'Enable', 'Off');
set(handles.stop_pushbutton, 'Enable', 'Off');
set(handles.clear_pushbutton, 'Enable', 'On');

guidata(hObject, handles);

function clear_pushbutton_Callback(hObject, eventdata, handles)

set(handles.start_pushbutton, 'Enable', 'On');
```

```
% Clears everything : both input and both axes
set(handles.maxParticles_input,'string','0');
set(handles.alpha_input,'string','0');
set(handles.seeds_input,'string','0');
set(handles.lattice_popmenu,'value',1)

cla(handles.axes1);

guidata(hObject, handles); %updates the handles

function boxcount_pushbutton_Callback(hObject, eventdata, handles)
c = getappdata(0, 'coor');
boxcount(c,'slope')

guidata(hObject, handles);
% ---------------------------- END BUTTONS ----------------------------
---


% ---------------------------- INPUTS ---------------------------------
---
function maxParticles_input_Callback(hObject, eventdata, handles)
%store the contents of input_alpha as a string. if the string
%is not a number then input will be empty
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input_alpha to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

function maxParticles_input_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function alpha_input_Callback(hObject, eventdata, handles)
%store the contents of input_alpha as a string. if the string
%is not a number then input will be empty
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input_alpha to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

function alpha_input_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function seeds_input_Callback(hObject, eventdata, handles)
%store the contents of input_alpha as a string. if the string
%is not a number then input will be empty
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input_alpha to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

function seeds_input_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if         ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function lattice_popmenu_Callback(hObject, eventdata, handles)

function lattice_popmenu_CreateFcn(hObject, eventdata, handles)

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if         ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% ---------------------------- END INPUTS ------------------------------
---
```